1992年12月1日発行（毎月1回1日発行）第11巻12号通巻128号 昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

Oh!X

Oh!X 5周年記念特別企画 ショートプロ大集合

各種ゲーム/地形表示プログラム/ランダムドット風立体視
Oh!XとOh!Xの読者の統計/モニタ募集/エレショウレポート
SX-WINDOW用追いかけっこゲーム/キャラグラゲームのススメ

12
1992

SOFT BANK オー！エックス
定価600円

SHARP

目の付けどころが、シャープでしょ。

"感性"咲かせるワー

POWER WORKSTATION

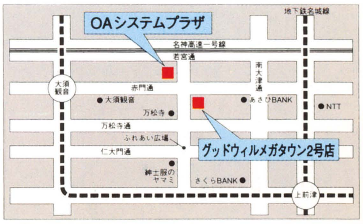
インテリジェントなパフォーマンスを誇るX68000Compact XVIと多彩にラインアップされたペリフェラル。感性を刺激するクリエイティブなワークステーション環境が自在に構築できます。

- パーソナルワークステーション(2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)
  CZ-674C-H(グレー) 標準価格**298,000**円(税別)
- 15型カラーディスプレイテレビ
  CZ-614D-TN(チタンブラック)・**-BK**(ブラック) 標準価格**135,000**円(税別)
  ■ディスプレイテレビ/CZ-6TU用RGBケーブル**CZ-6CR1** 標準価格**4,500**円(税別)
  ■ディスプレイテレビ/CZ-6TU用TVコントロールケーブル**CZ-6CT1** 標準価格**5,500**円(税別)
- 80MB内蔵用ハードディスクドライブ
  CZ-68HA 好評発売中
- 5.25インチ増設用フロッピーディスクドライブ
  CZ-6FD5 標準価格**99,800**円(税別・接続ケーブル同梱)
- 光磁気ディスクユニット
  CZ-6MO1 標準価格**450,000**円(税別)
  ■SCSI変換ケーブル**CZ-6CS1** 標準価格**12,000**円(税別)
- 2MB増設RAMボード
  CZ-6BE2D 標準価格**54,800**円(税別・取り付け費別)
  ■2MB増設RAM**CZ-6BE2B** 標準価格**54,800**円(税別・取り付け費別)×2
  ■数値演算プロセッサ**CZ-6BP2** 標準価格**45,800**円(税別・取り付け費別)
- 48ドット熱転写カラー漢字プリンタ
  CZ-8PC5-BK(ブラック) 標準価格**96,800**円(税別)
- MIDIボード
  CZ-6BM1A 標準価格**26,800**円(税別)
- インテリジェントコントローラ
  CZ-8NJ2 標準価格**23,800**円(税別)

感度抜群

キミのチャレンジを待つ!「X68000見・体・験フェア」開催

話題沸騰、ZOOMのF1新作ソフト<OVER TAKE>大ゲーム大会開催。腕に覚えのある方、ぜひご参加ください。
また「第1回全日本X68000芸術祭」全国大会エントリー作品もモチロンご覧になれます。
前回ご来場になれなかった方、今回はお見逃しなく!!



資料請求券
X68000ペリフェラル
Oh!X
12係

X68000
PERSONAL WORKSTATION・XVI
Compact

クステーション環境。

GRAPHIC WORKSTATION

- パーソナルワークステーション(2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)
  CZ-674C-H(グレー) 標準価格**298,000**円(税別)
- 21型カラーディスプレイ **CU-21HD** 標準価格**148,000**円(税別)
- 80MB内蔵用ハードディスクドライブ **CZ-68HA** 好評発売中
- 光磁気ディスクユニット **CZ-6MO1** 標準価格**450,000**円(税別)
  ■SCSI変換ケーブル**CZ-6CS1** 標準価格**12,000**円(税別)
- 2MB増設RAMボード **CZ-6BE2D** 標準価格**54,800**円(税別・取り付け費別)
  ■2MB増設RAM**CZ-6BE2B** 標準価格**54,800**円(税別・取り付け費別)×2
  ■数値演算プロセッサ**CZ-6BP2** 標準価格**45,800**円(税別・取り付け費別)
- カラーイメージスキャナ
  CZ-8NS1 標準価格**188,000**円(税別)
  ■スキャナ用パラレルボード**CZ-6BN1** 標準価格**29,800**円(税別)
- カラーイメージジェット
  IO-735X-B(ブラック) 標準価格**248,000**円(税別)
  ■接続ケーブル**IO-73CX** 標準価格**5,500**円(税別)

STANDARD WORKSTATION

- パーソナルワークステーション
  (2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)**CZ-674C-H**(グレー) 標準価格**298,000**円(税別)
- 14型カラーディスプレイ**CZ-608D-H**(グレー) 標準価格**94,800**円(税別)
- 5.25インチ増設用フロッピーディスクドライブ**CZ-6FD5** 標準価格**99,800**円(税別・接続ケーブル同梱)

TFT COLOR LCD WORKSTATION

- パーソナルワークステーション
  (2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)**CZ-674C-H**(グレー) 標準価格**298,000**円(税別)
- 10.4型カラー液晶ディスプレイ**LC-10C1-H**(グレー) 標準価格**598,000**円(税別)
  ■接続ケーブル**AN-1515X** 標準価格**4,200**円(税別)

※カラー液晶ディスプレイを接続してご使用の場合、SX-WINDOW上のアプリケーション利用に限定されます。

開催日時:12月5日(土)・6日(日)13:00〜17:00
「OVER TAKE」ゲーム大会 14:00〜16:00
会　場:(株)OAシステムプラザ大須店
名古屋市中区大須3-11-19(OAビル) ☎(052)265-1650
交通／名城線上前津駅下車徒歩3分
■主催:(株)OAシステムプラザ
■問い合わせ先:シャープエレクトロニクス販売(株)
中部統轄情報第2営業部 ☎(052)323-5145 担当・曽田

開催日時:12月12日(土)・13日(日)13:00〜17:00
「OVER TAKE」ゲーム大会 14:00〜16:00
会　場:(株)グッドウィル メガタウン2号店
名古屋市中区大須3-30-93(メガタウン1F) ☎(052)242-8581
交通／名城線上前津駅下車徒歩3分
■主催:(株)グッドウィル
■問い合わせ先:シャープエレクトロニクス販売(株)
中部統轄情報第2営業部 ☎(052)323-5145(代) 担当・森

■お問い合わせは…電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部AVCシステム事業推進室 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)
シャープ株式会社

Oh!X

CONTENTS

●Oh!X5周年特別企画

49 愛読者特別モニタ大募集

73 ショートプロ大集合

74 Where is Mistake? 金子俊一

76 IFF，ILBMのファイル情報を読む 影山裕昭

78 STRFIND.C 古村 聡

80 MAGICAL TRIANGLE 浜崎正哉

82 地形表示プログラム 中野修一

84 裸眼立体視（ランダムドット風） 丹 明彦

86 撃ち合いゲーム 石上達也

88 高次方程式のグラフを描く 横内威至

90 パワーダウンマネージャ 瀧 康史

93 あなたと私の電卓物語 西川善司

96 Oh!XとOh!Xの読者の統計 浦川博之

●カラー紹介

20 Oh!X Graphic Gallery
DōGA・CGアニメーション講座

21 SHOW REPORT
エレクトロニクスショウ'92

●THE SOFTOUCH

24 SOFT WARE INFORMATION
新作ソフトウェア/TOP10

26 TREND ANALYSIS

GAME REVIEW

28 デスブレイド 八重垣那智

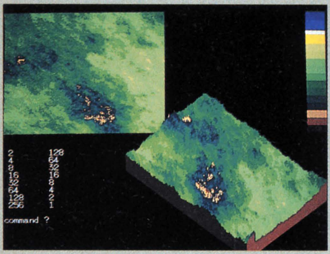
30 ふしぎの海のナディア 文月 涼

32 ロードス島戦記II 高橋哲史

34 ムーンクレスタ&テラクレスタ 西川善司

36 バーンウェルト 古村 聡

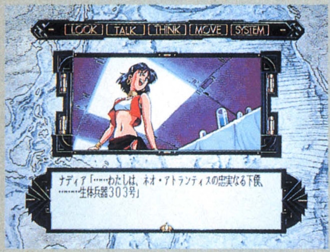
38 AFTER REVIEW
三國志III



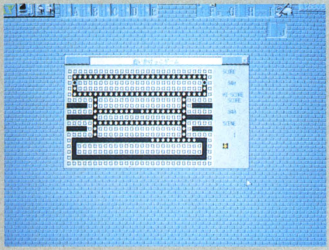
ショートプロ大集合

エレクトロニクスショウ'92

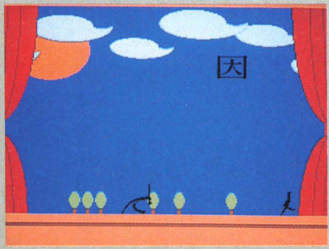
デスブレイド



ふしぎの海のナディア



SX-WINDOW用追いかけっこゲーム



シュールな風景

〈スタッフ〉
●編集長／前田　徹　●副編集長／植木章夫　●編集／浅井研二　山田純二　豊浦史子　●協力／有田隆也　中森　章　林　一樹　吉田幸一　華門真人　吉田賢司　影山裕昭　大和　哲　村田敏幸　丹　明彦　三沢和彦　長沢淳博　宮島　靖　金子俊一　浦川博之　石上達也　柴田　淳　御木徳高　瀧　康史　●カメラ／杉山和美　●イラスト／山田晴久　寺尾響子　高橋哲史　川原由唯　●アートディレクター／島村勝頼　●レイアウト／元木昌子　ADGREEN　●校正／グループごじら

表紙絵：須藤　牧人

E　N　T　S

1992 DEC.
12

●シリーズ全機種共通システム

137 THE SENTINEL

138 実践Small-C講座〔8〕MAKE　石上達也

●読みもの

148 第65回　知能機械概論ーお茶目な計算機たちー
アヴァン・ポップで仮想空間から逃げ出せ　有田隆也

150 猫とコンピュータ　第75回
肩コリと本棚　高沢恭子

152 X-OVER NIGHT　第29話
不良資産　高原秀己

●連載/紹介/講座/プログラム

22 響子 in CG わ〜るど［第19回］
思索カード　寺尾響子

40 DōGA CGアニメーション講座 ver. 2.50（第4回）
打倒TORNADOへの第一歩（完結編）　かまたゆたか

52 大人のためのX68000［第26回］
ついに発売，MATIER　荻窪　圭

54 ハードウェア工作入門〈30〉コンピュータアーキテクチャ編
レジスタ加算器の設計　三沢和彦

59 新製品紹介
版下作成支援ツールY-300A

60 SX-WINDOW対応
追いかけっこゲーム　石上達也

102 キャラグラゲームのススメ
シュールな風景　柴田　淳

107 X68000マシン語プログラミング Chapter_25H
探しもの　村田敏幸

119 Creative Computer Music入門(15)
弦のアレンジ　瀧　康史

125 OhIX LIVE in '92
LAST CHRISTMAS（X68000・Z-MUSIC用）　遠藤隆一
闇の血族・次回予告のテーマ（X68000・ZMUSIC.FNC用）　小野美樹夫
ユーフォリー・オープニングテーマ（X1・MusicBASIC用）　西尾将人

132 (で)のショートプロぱーてぃ　その39
散らかしOK，片づけOK　古村　聡

146 ANOTHER CG WORLD　寺尾響子

153 吾輩はX68000である［第18回］
極楽た〜ぼマウス　泉　大介

158 マシン語カクテル in Z80's Bar　第37回
ユーティリティがほしい　金子俊一

OhIX INDEX '92……162
ペンギン情報コーナー……166
FILES OhIX……168
OhIX質問箱……170
STUDIO X……172
編集室から/DRIVE ON/ごめんなさいのコーナー/SHIFT BREAK/microOdyssey……176

UNIXはAT & T BELL LABORATORIESのOS名です。
Machはカーネギーメロン大学のOS名です。
CP/M, P-CPM, CP/Mplus, CP/M-86 CP/M-68K, CP/M-8000, DR-DOSはデジタルリサーチ
OS/2はIBM
MS-DOS, MS-OS/2, XENIX, MACRO80, MS C, MS-WindowsはMICROSOFT
MSX-DOSはアスキー
OS-9, OS-9/68000, OS-9000, MW CはMICROWARE
UCSD p-systemはカリフォルニア大学理事会
TURBO PASCAL, TURBO C, SIDEKICKはBOLAND INTERNATIONAL
LSI CはLSI JAPAN
HuBASICはハドソンソフト
の商標です。その他，プログラム名,CPUは一般に各メーカーの登録商標です。本文中では"TM","R"マークは明記していません。
本誌に掲載されたプログラムの著作権はプログラム作成者に保留されています。著作権上，PDSと明記されたもの以外，個人で使用するほかの無断複製は禁じられています。

■広告目次

アイビット電子……183(上)
アクセス……184
金子製作所……12
カプコン……11
計測技研……181
J＆P……表3
シティソフト……183(下)
シャープ……表2・表4・1・4-9
九十九電機……13
ハミングバードソフト……19
P＆A……14・15
ブラザー工業……10
マイクロウェアシステムズ……182(上)
マイコンショップ川口……180
満開製作所……179

SHARP

目の付けどころが、シャープでしょ。

X68000 CompactXVI NEWS

Opinion 1

（ハードディスクが使いたい。）

Compact専用の内蔵ハードディスクが登場します。SCSI仕様の80MB。場所を取らずに高速・大容量ファイル環境を実現します。

■内蔵用ハードディスクドライブ（CZ-674C専用）
CZ-68HA……………………好評発売中
※取りつけに関してはシャープお客様ご相談窓口にてご相談ください（取りつけ費別）。

さらに大容量をお望みの場合、外付け用のSCSI端子で一般のSCSIハードディスクも接続可能。フルピッチSCSI端子とハーフピッチSCSI端子を接続するためのSCSI変換ケーブルも用意しています。

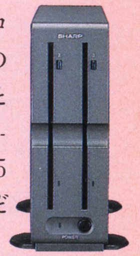
■SCSI変換ケーブル
CZ-6CS1………標準価格12,000円（税別）

CZ-6CS1

Opinion 2

（従来のソフト資産を活かしたい。）

これについても、Compact専用の外付け5インチフロッピーディスクユニットを用意していますから、従来の68シリーズの資産を有効活用できます。3.5インチと5インチの間でのデータのやりとりも可能。また、CZ-674C及びCZ-6FD5のスイッチ設定を変えれば、5インチソフトからの起動が可能になり、市販ソフトなどそのまま使えます。

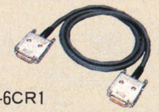


■増設用5インチ・フロッピーディスク・ユニット（CZ-674C専用）
CZ-6FD5……………………標準価格99,800円（税別）

Opinion 3

（ディスプレイテレビを接続したい。）

Compactは、従来のシリーズと比べ体積比44%と小さいため、コネクタの形状も異なっていますが、このケーブルを使用することにより、ディスプレイテレビやRGBシステムチューナーを利用できます。



CZ-6CR1

CZ-6CT1

■15型カラーディスプレイテレビ（スピーカー・チルトスタンド同梱）
CZ-614D-TN…………標準価格135,000円（税別）

■ディスプレイテレビ/CZ-6TU用RGBケーブル
CZ-6CR1……………………標準価格　4,500円（税別）

■ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル
CZ-6CT1……………………標準価格　5,500円（税別）

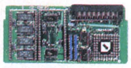
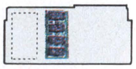
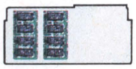
資料請求券
X68000ペリフェラル
Oh!X
12係

パーソナルワークステーション X68000 Compact XVIについてのご意見、ご要望にお応えします。

Opinion 4

（ メモリ環境をパワーアップしたい。 ）

Compactは2MBのメインメモリを標準装備していますが、本体内で最大8MBまで拡張できます。

	容量	周辺機器
標準	2MB	——
拡張	4MB	 CZ-6BE2D
	6MB	 CZ-6BE2B
	8MB	 CZ-6BE2B×2

■2MB増設RAMボード CZ-6BE2D 標準価格54,800円（税別）

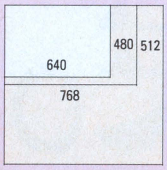
■2MB増設RAM CZ-6BE2B 標準価格54,800円（税別）

※取りつけに関してはシャープお客様ご相談窓口にてご相談ください（取りつけ費別）。

Opinion 5

（ 液晶ディスプレイとSX-WINDOWの関係は？ ）

液晶ディスプレイ（LC-10C1-H 標準価格598,000円・税別）の解像度は640×480ドット。Compactでは、従来のX68000シリーズの画面モードにこの画面モードをプラス。解像度の制約を受けないウィンドウ環境ならではの機能です。このようにSX-WINDOW環境の確立により、ハードウェアに依存しない快適な操作環境が実現します。



Opinion 6

（数値演算プロセッサはほんとに速い？）

ご存じのようにMPU68000自体は複雑な計算（浮動小数点演算）を単純な計算の組み合わせで行っています。X68000シリーズに装備されている浮動小数点演算パッケージ「FLOAT2.X」は、よく使う単純な組み合わせをまとめたもの。数値演算プロセッサは、いわばこのパッケージの機能を、ハードウェアで高速に実現し、MPUの負担を軽くするものです。アプリケーションプログラムの中には浮動小数点演算を必要としないものもあるため、すべてのプログラムが高速になるわけではありませんが、レイトレーシングなど大量の実数演算を必要とするソフトウェアの場合、飛躍的な実行速度の向上が期待できます。

■数値演算プロセッサ CZ-6BP2 標準価格45,800円（税別）

※数値演算プロセッサはCZ-6BE2D上に装着します。
※取りつけに関してはシャープお客様ご相談窓口にてご相談ください（取りつけ費別）。

X68000
PERSONAL WORKSTATION・XVI
Compact

本体＋キーボード＋マウス
2HD3.5インチFDDタイプ CZ-674C-H（グレー） 標準価格298,000円（税別）
14型カラーディスプレイ（ドットピッチ0.28mm）
CZ-608D-H（グレー） 標準価格94,800円（税別）

■お問い合わせは…電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06) 621-1221（大代表）
電子機器事業本部AVCシステム事業推進室 〒162 東京都新宿区市谷八幡町８番地 ☎(03)3260-1161（大代表）

シャープ株式会社

SHARP

カラープリンタもスキャナも………

黒の統一美。

画像処理のベストマッチングシステム for X68000。

シャープ株式会社　〒545 大阪市阿倍野区長池町22番22号　☎(06)621-1221(大代表)

入力から、編集、出力、複写まで、カラーの世界は、シャープです。

SHARP IS COLOR

▶INPUT

X68000用パラレルインタフェースを標準装備した高速コンパクト型イメージスキャナ。

カラーイメージスキャナ JX-220X……標準価格168,000円(税別)

●A4サイズの原稿を約50秒※1で高速読み取り●CCDセンサー採用。さらに中間調処理でシャープでリアルな画像を再現●ディザパターン指定機能※2や濃度補正機能※2など高度な画像処理機能で緻密な読み取りが可能●解像度200ドット/インチ(約7.9ドット/㎜)。ズーム機能で1%きざみの拡大、縮小も可能●色ずれの少ない線順次(1走査)読み取り●X68000シリーズ用「スキャナツール」ソフトを標準装備●プリンタと直接接続することによりダイレクトプリント※3が可能●RS-232Cインタフェース/X68000シリーズ用専用パラレルインタフェースを標準装備。

※1:A4、2値出力、コンピュータへの実転送時間。
※2:表記機能はJX-220X本体使用であり、付属ユーティリティ使用時は異なります。
※3:別売のパラレルインタフェースケーブル(JX-22PC標準価格12,000円(税別))が必要です。

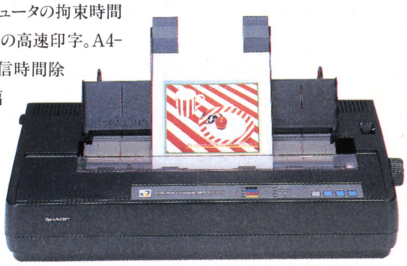
▶OUTPUT

3種類の制御コマンドモードを搭載。質感も鮮やかに再現する高品位カラーイメージジェット。

カラーイメージジェット IO-735X-B……標準価格248,000円(税別)

●シャープ独自のIOシリーズコマンド(Gモード)に加え、NM-9900モード(Nモード)、ESC/P24-84C準拠モード(Pモード)をサポート。一般文書の作成から、各種デザイン、建築用パースなどのCAD分野に対応●発色性に優れた普通紙対応の新黒インキ採用。専用紙はもちろんオフィスでよく使われる普通紙にも鮮明カラー印字●プリントバッファメモリ(128KB)の内蔵で、ホストコンピュータの拘束時間を軽減●48ノズル(各色12ノズル)採用の高速印字。A4-1ページを※約90秒でプリント(データ受信時間除く)●ビジネス用途に適したB4横用紙幅対応●OHPフィルム(専用)にも鮮明プリント●ノンインパクト方式ならではの静粛印字●インキ補充は簡単、経済的なカートリッジ方式

※261×174㎜領域



IO-735X-B 対応アプリケーション

●SX-WINDOW対応ペイントツール
Easypaint SX-68K
CZ-263GW 標準価格12,800円(税別)

●WYSIWYGを実現、ドローグラフィックソフト
CANVAS PRO-68K
CZ-249GS 標準価格29,800円(税別)

●オリジナリティを活かせるポップアップツール
NEW Printshop PRO-68K ver2.0
CZ-221HS 標準価格20,000円(税別)

●マルチワープロ PRO-68K
Multiword
CZ-225BS 標準価格32,000円(税別)

●高速カード型リレーショナルデータベース
CARD PRO-68K ver2.0
CZ-253BS 標準価格29,800円(税別)

●パソコン通信もできるメモリ常駐型ソフト
Teleportion PRO-68K
CZ-258BS 標準価格22,800円(税別)

●これからの高速通信をサポート
Communication PRO-68K ver2.0
CZ-257CS 標準価格19,800円(税別)

資料のご請求・お問い合わせはコンシューマーセンター
●東日本相談室 〒261 千葉市美浜区中瀬1丁目9番2号 ☎(043)297-1221(大代表) ●西日本相談室 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

SHARP

目の付けどころが、シャープでしょ。

X68000
PERSONAL WORKSTATION・XVI
Compact

本体＋キーボード＋マウス
2HD3.5インチFDDタイプ
CZ-674C-H（グレー） 標準価格298,000円（税別）
14型カラーディスプレイ（ドットピッチ0.28mm）
CZ-608D-H（グレー） 標準価格94,800円（税別）

●5.25インチ増設用フロッピーディスクドライブ
CZ-6FD5
標準価格99,800円・税別
〔接続ケーブル同梱〕

●ディスプレイテレビ/CZ-6TU用RGBケーブル
CZ-6CR1 標準価格4,500円・税別
●ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル
CZ-6CT1 標準価格5,500円・税別
●SCSI変換ケーブル CZ-6CS1 標準価格12,000円・税別

待望のSX-WINDOW開発支援ツール、登場。

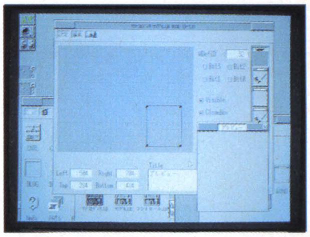
SX-WINDOW 開発キット Workroom SX-68K

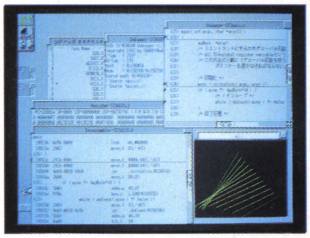
NEW

CZ-288LWD 12月発売予定

SX-WINDOW用のソフト開発に必要な開発ツールやサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます。初めてSX-WINDOW用のプログラムに挑戦する人にも、簡単に基本機能の理解ができる33種のサンプルプログラム付き。また各マネージャ解説と関数リファレンスの詳細なマニュアルも装備しています。

※本ソフトのご使用に際しては、メインメモリ4MB以上、SX-WINDOW ver2.0以上、C compiler PRO-68K ver 2.1以上が必要です。





キット構成

■開発ツール

●SXデバッガ
SX-WINDOW上で複数のプログラムを同時にデバッグすることができるソースコードデバッガ。

●リソースエディタ
SX-WINDOW上のリソースをリソースタイプごとの編集ウィンドウでビジュアルに作成・編集が可能。

●リソースリンカ
Cコンパイラやアセンブラで作成したリソースデータファイル（オブジェクトファイル）をリンクしてリソースファイルを作成。

●サンプルメイク
サンプルプログラムのコンパイル作業をSX-WINDOW上から、XC ver2のMAKE.Xを呼び出して、自動実行する簡易メイクユーティリティ。

■サンプルプログラム

●基礎編（23種）
各マネージャの基本的な機能のみを用いた基本動作の理解。

●応用編（4種）
基礎編での基本機能を応用した簡単なアプリケーションの作成。

●実用編（6種）
基礎/応用編での機能を駆使した、実用的なアプリケーションの作成。

■その他のファイル

●インクルードファイル
Cコンパイラとアセンブラ用の関数定義、データ定義ファイル。

●ライブラリファイル
Cコンパイラ用の関数ライブラリ。

マニュアル
●ユーザーズマニュアル ●プログラマーズマニュアル ●ファンクションリファレンス ●ライブラリリファレンス

資料請求券
CZ/ソフト
Oh!X
12係

開いてくださいウィンドウ、触れてくださいインテリジェンス。
さらに拡がる、SXワールド。

●アウトラインフォント対応、ひらかれたウィンドウ環境。

SX-WINDOW ver2.0

CZ-287SS　標準価格12,800円(税別)

フォントマネージャを装備して待望のアウトラインフォントに対応。画面スクロール機能により、表示画面よりワイドなデスクトップ空間を駆使。アプリケーションのハンドリングに便利なシンボルトレイやアイコンメンテ、パターンエディタなど便利機能満載。

※SX-WINDOW ver1.0(CZ-259SS)およびSX-WINDOW ver1.1(CZ-278SS)をお持ちの方には有償バージョンアップを行います。

●マルチタスク機能をはじめ、通信環境がさらに充実。

Communication SX-68K

NEW

CZ-272CWD　標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションソフトを実行中でも簡単に通信が可能。また、ホスト局をクリックするだけの自動ログイン機能、初心者にも簡単なプログラム機能、最新モデム(20種類)もフルサポートしています。

●多彩なサウンドクリエイトを実現するFM音源サウンドエディタ。

SOUND SX-68K

NEW

CZ-275MWD　標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成・変更ができるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。まさにミキサー感覚で音創りが楽しめるツールです。

●ウィンドウ対応グラフィックツール。

Easypaint SX-68K

CZ-263GWD　標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。

※SX-WINDOW対応ソフトの動作には、メインメモリ2MBおよびSX-WINDOW ver1.1以上が必要です。

充実のPROシリーズ

●ビジネスグラフチャート

CHART PRO-68K

NEW

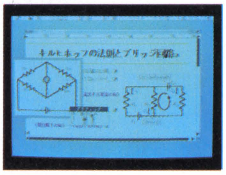
CZ-267BSD　標準価格38,000円(税別)

各種データベースで作成したデータをもとに、多彩なグラフが作成できます。3次元表示やグラフの複合機能も装備。データはMultiword,Press Conductor PRO-68Kに取り込むこともできます。

●グラフィック機能搭載の本格派ワープロ

Multiword ver1.1

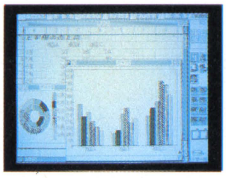
CZ-225BSD　標準価格32,000円(税別)



●簡単操作の統合型表計算ソフト

BUSINESS PRO-68K Popular

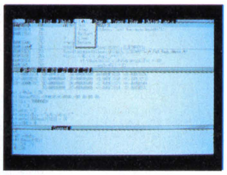
CZ-286BSD　標準価格28,000円(税別)



●各種ドライバ、ライブラリを追加

COMPILER ver2.1 PRO-68K

CZ-285LSD　標準価格44,800円(税別)

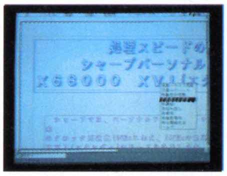


※有償バージョンアップ対応中。

●各種エディタ装備のレイアウトソフト

PressConductor PRO-68K

CZ-266BSD　標準価格28,000円(税別)



※以上のPROシリーズのソフトの動作にはメインメモリ2MB必要です。

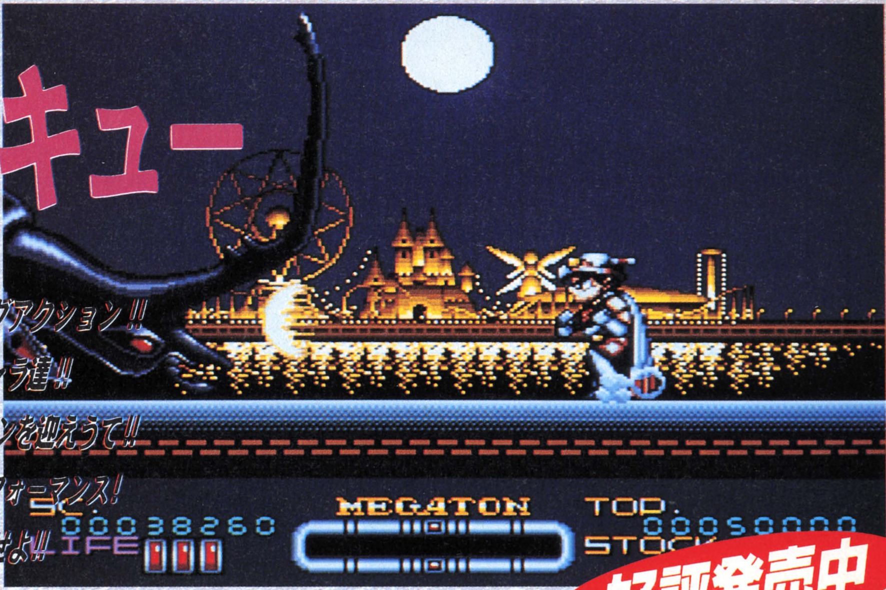
※発売予定のソフトの画面写真は実物とは異なる場合があります。

●お問い合わせは…シャープ㈱電子機器事業本部AVCシステム事業推進室　〒162 東京都新宿区市谷八幡町8番地　☎(03)3260-1161(大代表)へ。シャープ株式会社

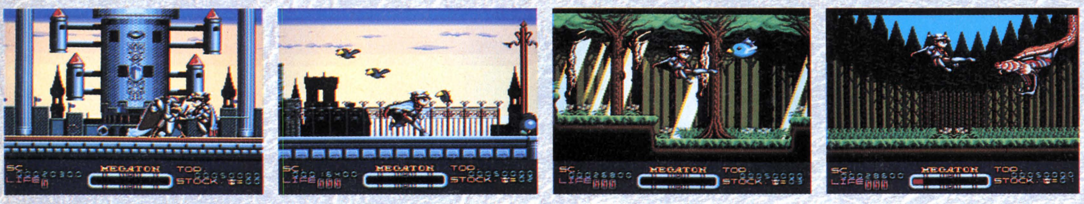
brother

パソコンソフト自販機
TAKERU

サンダーレスキュー



- 全7面のステージ！横スクロール・シューティングアクション!!
- 3重スクロールの美しいグラフィックと巨大ボスキャラ達!!
- スムーズ&パワフルな動き！多彩な敵の攻撃パターンを迎えうて!!
- 難易度高め。何度でもトライできる、ハイコストパフォーマンス!
- ステージは、マルチチョイス、得意なボスから攻略せよ!!



好評発売中

X68000版

TAKERU価格 ¥4,800(税込)

■企画/開発：ギミックハウス

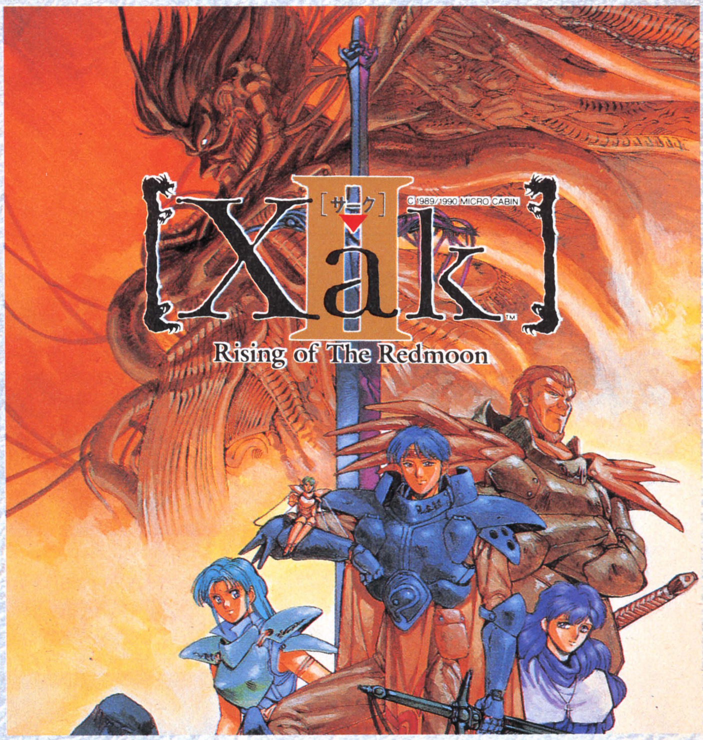
チェイスH.Q.

CHASE H.Q.

好評発売中

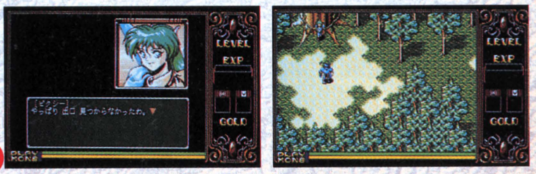
TAKERU価格
¥7,800(税込)

©TAITO



TAKERU価格
¥6,800(税込)

©1989/1990
マイクロキャビン



発売中

ブラザー工業株式会社 〒467 名古屋市瑞穂区苗代町2番1号

TAKERU事務局
(052)824-2493

東京営業所(03)5203-7133
大阪営業所(06)252-4234

通信販売

通信販売をご希望の方は、ソフト名・機種名・住所・氏名・電話番号を明記の上TAKERU事務局まで現金書留でお申し込み下さい。
代金引換は一度、現金書留で申し込んで頂いた方に案内させて頂きます

CAPCOM
炸裂する超スタントアクション。
幻の飛竜、いまX68000に登場!
©CAPCOM 1989 ALL RIGHTS RESERVED.
CAPCOM X68000 ソフト 第2弾
ストライダー飛竜®
必殺技・サイファが炸裂する!
斬る、跳ぶ、宙返る、滑り込む、駆けあがる…。すべてが人間の能力をはるかに超えた、
ダイナミック&スピーディなスタントアクションの数々!!
さらに、緻密に構成された各ステージと抜群の操作性!
あの興奮を搭載し、いま幻の飛竜がX68000で甦る―。
ジョイスティック対応
No Copy このマークは不法コピー禁止マークです
ソフトウエア法的保護監視機構
11月27日発売予定 定価9,800円(税別)
株式会社カプコン 大阪本社営業 〒540 大阪市中央区釣鐘町2−2−8 東京支店 〒163-02 東京都新宿区西新宿2−6−1新宿住友ビル43F
★カプコンソフト情報★ 大阪(06)946-6659 東京(03)3340-0718 札幌(011)281-8834 仙台(022)214-6040 名古屋(052)571-0493
広島(082)243-6264 松山(0899)34-8786 福岡(092)441-1991
電話番号は、よく確かめておかけ間違いのない様にしてください

KANEKO
究極TIGER
激戦ふたたび。
X68000
5"2HD(2枚組)
[redacted]
¥8,800(税別)
12月発売予定
© 1992 KANEKO Co., Ltd.
KANEKO CO., LTD.
株式会社 金子製作所 〒177 東京都練馬区石神井台8丁目23番地21号 TEL.03(3921)9661

親切と安さの TSUKUMO　掲載商品2万円以上 送料無料(一部地域を除く)

SHARP わんさかフェアー

TSUKUMO 30TH YEAR SINCE 1962

秋葉原電気まつり
4年11月20日(金)〜5年1月7日(木)
賞金総額6,000万円
★1等 現金10万円 ★2等 現金5万円
★3等 現金1万円 ★4等 現金5千円
※店頭にて5,000円お買い上げ毎に1枚、抽選券をプレゼント!

シャープX68000の事なら何でも揃うツクモにおまかせ!

秋葉原を歩き回る必要はありません。情報が沢山。分らない事は何でもお尋ね下さい。目に優しい**10.4型カラー液晶ディスプレイ**(LC-10CI)も取り扱い中!詳しくはお問い合わせ下さい。システムのご相談は☎03(3253)1899までどうぞ。

恒例

「SHARPわんさかフェアー in TSUKUMO」開催

12月5日(土)・6日(日) ツクモパソコン本店2Fにて

- グラフィックツール「マチエール」の使いこなし
- MIDIソフト「Mu-1 Super」の実演
- 特価品大特売!!(何があるかはお楽しみ)

※X68000の使い方を中心とした内容のデモと特価品を取り揃えてお待ちしております

X68000いろいろ組み合わせ提案いたします。

- X68000の未来を象徴するハイコンパクトなボディ(体積比44%)
- 成熟するウィンドウ環境、使いやすさと高機能を追求したSX-WINDOW Ver2.0搭載
- 2HD 3.5インチフロッピーディスクドライブ2基搭載
- カラー液晶ディスプレイ接続可能
- X68000XVIの高性能を継承
- VGAモードサポート(SX-WINDOWのみ対応)

●5インチソフトも使える欲張りセット
CZ-674C-H…定価¥298,000
X68000 Compact本体
CZ-608D-H…定価¥94,800
0.28mmピッチカラーディスプレイ
5インチドライブ(2ドライブ)サービス
ツクモ特価¥336,000

●ハードディスクで便利に使えるセット
CZ-674C-H…定価¥298,000
X68000 Compact本体
CZ-608D-H…定価¥94,800
0.28mmピッチカラーディスプレイ
100MBハードディスク サービス
ツクモ特価¥348,000

ツクモオリジナル

目のつけどころがツクモでしょ

■X68000シリーズ専用3.5インチフロッピーディスクドライブ
TS-3XRシリーズ
- 3.5インチ2DD/2HD/2HC/1.44MBフォーマット対応(いろいろなフォーマットのメディア読み書きが出来ます。
- ユーティリティソフト付属(デバイスドライバー/フォーマッター)

TS-3XR1(1ドライブ)
定価¥44,800 **ツクモ特価¥35,800**
TS-3XR2(2ドライブ)
定価¥57,800 **ツクモ特価¥46,800**

※写真はTS-3XR1です。

■X68000 Compact XVIシリーズ用5インチフロッピーディスクドライブ
TS-5XRシリーズ NEW
- 5インチ2HD/2DDフォーマット対応
- ドライブ番号切り換えスイッチ付

TS-5XR1(1ドライブ) 定価¥53,800 **ツクモ特価¥42,800**
TS-5XR2(2ドライブ) 定価¥72,800 **ツクモ特価¥57,800**

耳よりな情報—X68000XVI/Compact XVIシリーズお持ちの方

NEW
2MB増設RAM
(CZ-6BE2Bコンパチ)
既に、内蔵メモリーボードを搭載して4MBに増設されている方で、更に増設をお考えの方へお勧め商品
TS-6BE2B この冬特別限定生産 **ツクモ特価¥34,800**

SCSIタイプハードディスク

VIP 100CX(100MB ダークグレー) **ツクモ特価¥68,000**
VIP 120CX(120MB ダークグレー) **ツクモ特価¥78,000**
LHD-FM200E(200MB) **ツクモ特価¥98,000**
LHD-FM240(240MB) **ツクモ特価¥128,000**
※SCSIボード(CZ-6BS1 定価¥29,800)は別売です。

ラストチャンス!!
SASIタイプハードディスク 限定
SHD-40J
ツクモ特価¥49,800

超低金利/冬・夏ボーナス二回払受付中!!詳しくは各店までお問い合わせ下さい。

MIDIコンピュータミュージック特選セット

特選Aセット
- SC-55……¥69,000
- SX-68M-II……¥19,800
- Mu-1 SUPER……¥39,800

合計定価 ¥128,600
ツクモ特価¥99,000
(消費税別途¥2,970)
クレジット例(18回払・税込)
初回¥6,596+月々¥6,300×17回

特選Bセット
- CM-300……¥58,000
- SX-68M-II……¥19,800
- Mu-1 SUPER……¥39,800

合計定価 ¥117,600
ツクモ特価¥92,000
(消費税別途¥2,760)
クレジット例(10回払・税込)
初回¥10,919×月々¥10,000×9回

特選Cセット
- CM-500……¥115,000
- SX-68M-II……¥19,800
- Mu-1 SUPER……¥39,800

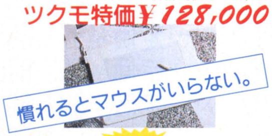
合計定価 ¥174,600
ツクモ特価¥141,000
(消費税別途¥4,230)
クレジット例(15回払・税込)
初回¥11,300+月々¥10,500×14回

スーパーグラフィックセット

その1
WACOM製
- SD-510C タブレット……¥98,000
- TJ-410A-2 接続ケーブル……¥6,000
- SP-200A スタイラスペン……¥10,000

サンワード
- Matier(マチエール)……¥39,800

合計定価¥ 153,800
ツクモ特価¥128,000



その2
ヒューレットパッカード HP Desk Jet 505J
インクジェットプリンタ……¥99,800
カラーキット……¥12,000
アーベル プリンターケーブル……¥4,800
サンワード Matier(マチエール)……¥39,800
合計定価 ¥156,400
ツクモ特価¥123,000

● JX-220X
A4サイズカラーイメージスキャナ‥定価¥168,000

メモリーボード

■1MB増設RAMボード(CZ-600C専用)
ツクモ特価¥19,500
■1MB増設RAMボード
(ACE PRO PRO2シリーズ用)
ツクモ特価¥17,000
■2MB増設RAMボード(拡張スロット用)
ツクモ特価¥33,800
■4MB増設RAMボード(拡張スロット用)
ツクモ特価¥59,800
※計測技研のメモリーボードも取り扱っております。価格についてはお尋ね下さい。

SX-WINDOWワールド

- SX-WINDOW開発キット
  CZ-288LWD 12月発売予定
- SX-WINDOW Ver.2.0
  CZ-287SS ¥12,800
- Communication SX-68K
  CZ-272CWD ¥19,800
- SOUND SX-68K
  CZ-275MWD ¥19,800
- Easypaint SX-68K
  CZ-263GWD ¥12,800

電子文具

タイムマネージメントを管理する便利ツール
●従来の電子システム手帳用ICカードがそのまま使えます●次から次へと忙しい方の為の強力な助っ人●大画面・大容量・手書き入力で操作効率向上!

シャープ 電子マネージメント手帳
PV-F1 定価¥128,000

ツクモ特価販売中!

更に、便利な名刺読み取り機「PV-BR1」(標準価格¥120,000)もお勧めです。

パソコン通信

時代は9600ボーへ!!
■モデム 9600bps MNP5 & CCITT V.42bis **ツクモ特価¥49,800〜**
■通信ソフト た〜みのる2
ツクモ特価¥14,000

液晶ビジョン

あなたの部屋がミニシアター&迫力ゲームセンターに変身!
シャープ液晶ビジョンセット
XV-P1 定価¥220,000

今なら
RGB信号→
S端子変換ユニット
プレゼント

ツクモ特価¥198,000

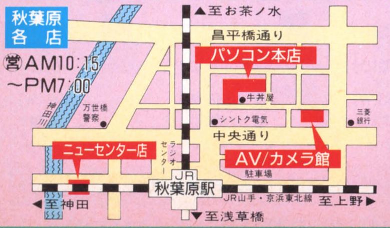
X68000用MOディスク

ツクモはSONY MOディスクの正規代理店です。
これが今一番の人気者!
SONY 3.5インチ光磁気ディスクユニットセット
- RMO-S350(3.5″光磁気ディスクドライブ) ¥235,000
- SCSIケーブル……¥6,900
- SCSIインターフェースボード……¥29,800

合計定価 ¥271,700
ツクモ特価販売中

通信販売のご注文は下記フリーダイヤルへ。
全国どこからでも通話料無料
受・注・専・用 フリーダイヤル **0120-377-999**
通販センター **03-3251-9911**　商品についてのお問い合わせは各店又は通販へ。

クレジット払い 月々¥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中!	**各種リース払い** くわしくは各店にお問い合わせ下さい。ケースに合わせてご相談にのります!	**現金書留払い** 〒101-91 東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh!X係
カード払い 通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通販部へお申し込み下さい。	**全国代金引き換え配達** お申し込みは☎03-3251-9911へお電話1本! 配達日の指定もできます。	**銀行振込払い** 事前に☎でお届け先をご連絡下さい。 三和銀行 秋葉原支店(普)1009939 ツクモデンキ



ツクモは「スーパーX PRO SHOP」です。

PRO STAFF **ツクモ**

九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号
★商品のご注文は在庫確認の上お願いします。★表示価格には消費税は含まれておりません。

ツクモパソコン本店2F
☎03-3253-1899(直通)
パソコン本店代表 ☎03-3253-5599 (担当/荒井) 休毎週木曜

■ツクモニューセンター店 ☎03-3251-0987(担当/沢栄) 休毎週木曜
■名古屋1号店 ☎052-263-1655(担当/山口) 休毎週火曜
■名古屋2号店 ☎052-251-3399(担当/松原) 休毎週水曜
■ツクモ札幌店 ☎011-241-2299(担当/田口) 休毎週木曜
■DEPOツクモ2番街店 ☎011-242-3199(担当/鈴木) 休毎週木曜

※12月中は、無休で営業致します。

P&Aならではの新品パソコン5年保証

《業界No.1の"P&Aメンテナンスサポート》

最高の保証システム

①業界最長の新品パソコン5年保証
（※モニター・プリンター３年間保証.// ※一部商品は除きます。）
②中古パソコンの１年間保証
（モニター・プリンター６ヶ月間保証）
③初期不良交換期間3ケ月
（※新品商品に限らせていただきます）
④永久買取保証
⑤配達の指定OK//
⑥夜間配送もOK//
（※PM6:00〜PM8:00の間 ※一部地域は除きます。）

便利でお得な支払いシステム

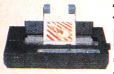
①翌月一括払い手数料無料（ご利用下さい。）
②業界No.1の低金利
③月々の支払いは¥1,000より
④9ヶ月先からのスキップ払いOK//
⑤84回までの分割、ボーナス併用OK//
⑥カレッジクレジット
⑦ステップアップクレジット
⑧ボーナスだけで10回払いOK//
⑨現金一括払いOK//
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

またまたまた

11/18〜12/17

《増設メモリー&数値演算プロセッサ》計測技研

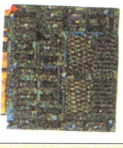
1 PRKII-02(2M)……定価¥ 55,000▶特価¥ 39,800
2 PRKII-04(4M)……定価¥ 90,000▶特価¥ 67,000
3 PRKII-06(6M)……定価¥125,000▶特価¥ 92,500
4 PRKII-08(8M)……定価¥160,000▶特価¥119,000
5 PRKII-12(2M)……定価¥ 85,000▶特価¥ 63,000
6 PRKII-14(4M)……定価¥120,000▶特価¥ 89,500
7 PRKII-16(6M)……定価¥155,000▶特価¥114,500
8 PRKII-18(8M)……定価¥190,000▶特価¥141,000
9 MC-68881RC……定価¥ 38,000▶特価¥ 27,000



カラーイメージジェット
■IO-735X-B
定価¥248,000
特価¥152,000
（送料・消費税込み¥157,590）

FDD（5インチ×2基）
■CZ-6FD5
（シャープ）（定価¥99,800）
P&A超特価.//
TEL下さい。

X68000メモリボード



①SH-6BE1-1M(600C専用)(I/Oデータ)定価¥25,000
（送料・消費税込み¥18,952）……特価¥17,900
②1MB増設RAMボード(ACE/PRO/PROII用)定価¥25,000
（送料・消費税込み¥16,583）……特価¥15,600
③2MB増設RAMボード(拡張スロット用)定価¥50,000
（送料・消費税込み¥32,239）……特価¥30,800
④4MB増設RAMボード(拡張スロット用)定価¥88,000
（送料・消費税込み¥55,620）……特価¥53,500

■Z,s STAFF PRO 68K Ver3.0
（ツアイト）（定価¥58,000）
特価¥37,500
（送料・消費税込み¥39,140）

■SX-68M II MIDI
（システムサコム）（定価¥19,800）
特価¥13,500
（送料・消費税込み¥14,420）

■CZ-68HA
●674C用内蔵HD80M
特価~~¥95,000~~
TEL下さい.//

注目//平成5年3月末一括払い手数料（金利）無料（平成4年12月末／平成5年1月末／2月末／3月末のいずれかをご指定下さい。）

X68000 CompactXVI/XVI/XVI-HD

※送料¥2,000、消費税別

今月の特選// 特価品

■Compact XVI さらにお安くなります。

●CZ-674C-H
●CZ-608D-H
●CZ-6FD5（5"FDD）
定価¥492,600
P&A超特価~~¥320,000~~
（※X68000サービスゲーム全て付いています。）
（モニターをCZ-606Dに変更の場合¥10,000を引いて下さい）

右記セットでお買い上げの方にもれなくプレゼント//
①「ダウンタウン熱血物語（¥8,800）」
はもちろん、さらにその上、人気の
㋑「ロードス島戦記（¥9,800）」
㋺「グラディウスII（¥9,800）」
㋩「ザ・プロサッカー68（¥9,800）」
㋥「信長の野望武将風雲録（¥9,800）」
㋭「ELLE(エル)（¥7,800）」
の中のいずれか2本をプレゼント//

X68000-CompactXVI

●ディスケット10枚●ジョイカード2ケプレゼント中// さらにお安くなります。

Ⓐセット：CZ-674C＋CZ-608D……定価¥392,800▶特価~~¥281,000~~

12回	23,400	24回	12,400	36回	8,600	48回	6,700	60回	5,600

X68000-XVI

▶セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中//

Ⓐセット：CZ-634C-TN＋CZ-606D-TN……定価¥447,800▶特価価格はTEL下さい。

12回	25,900	24回	13,700	36回	9,500	48回	7,500	60回	6,300

Ⓑセット：CZ-634C-TN＋CZ-614D-TN……定価¥503,000▶特価価格はTEL下さい。

12回	29,400	24回	15,500	36回	10,800	48回	8,400	60回	7,100

X68000-XVI-HD

▶セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中//

Ⓐセット：CZ-644C-TN＋CZ-606D-TN……定価¥597,800▶特価価格はTEL下さい。

12回	35,700	24回	18,900	36回	13,100	48回	10,300	60回	8,600

Ⓑセット：CZ-644C-TN＋CZ-614D-TN……定価¥653,000▶特価価格はTEL下さい。

12回	39,100	24回	20,700	36回	14,300	48回	11,200	60回	9,400

※上記のモニターを、CZ-606D（定価¥79,800）、CZ-604D（定価¥94,800）、CZ-607D（定価¥99,800）、CZ-605D（定価¥115,000）、CZ-608D（定価¥94,800）、CZ-614D（定価¥135,000）、CU-21HD（定価¥148,000）に変更の場合、TEL下さい。超特価で販売致します。

X68000シリーズ〜P&Aスペシャルセット

（送料¥2,000・消費税別）

SUPER-HD
（CZ-623C-TN）
●ハードディスク81MB搭載
●平均アクセスタイム19ms
●SCSIインターフェイス標準装備
●SX-WINDOW Ver.1.0搭載
●メインメモリ 2MB標準

SUPER-HD P&A特選セット ★ハードディスク81MB搭載//

Ⓐセット：■CZ-623C-TN（単品）…………定価¥498,000▶特価¥178,000
Ⓑセット：■CZ-623C-TN＋CZ-606D……定価¥577,800▶特価¥233,000
Ⓒセット：■CZ-623C-TN＋CZ-608D……定価¥592,800▶特価¥246,000
Ⓓセット：■CZ-623C-TN＋CZ-607D……定価¥597,800▶特価¥248,000
Ⓔセット：■CZ-623C-TN＋CZ-614D……定価¥633,000▶特価¥268,000
Ⓕセット：■CZ-623C-TN＋CU-21HD……定価¥646,000▶特価¥278,000

注目 スペシャルプレゼント//
★SUPER-HDには、
上記の①をプレゼント
★PRO-IIには、上記の
①＋㋑〜㋭の中の２本をプレゼント
ズバリ価格で大奉仕中
●ディスケット10枚、●ジョイカード2個プレゼント中

PRO-II P&A特選セット 限定

Ⓐセット：■CZ-653C（単品）……定価¥285,000▶特価¥138,000
Ⓑセット：■CZ-653C＋CZ-606D……定価¥364,800▶特価¥195,000
Ⓒセット：■CZ-653C＋CZ-604D……定価¥379,800▶特価¥197,000
Ⓓセット：■CZ-653C＋CZ-608D……定価¥379,800▶特価¥207,000
Ⓔセット：■CZ-653C＋CZ-607D……定価¥384,800▶特価¥209,000
Ⓕセット：■CZ-653C＋CZ-614D……定価¥420,000▶特価¥229,000
Ⓖセット：■CZ-653C＋CU-21HD……定価¥433,000▶特価¥239,000

X68000用ハードディスク（送料¥1,000 消費税別）

〈ロジテック〉
①LHD-FM100E（定価¥99,800）
▶P&A超特価TEL下さい。
②LHD-FM200E（定価¥138,000）
▶P&A超特価TEL下さい。
〈エニックス〉
③EFX-100（定価¥118,000）
▶P&A超特価TEL下さい。
④EFX-140（定価¥138,000）
▶P&A超特価TEL下さい。
〈ジェフ〉
⑤GF-120（定価¥108,000）
▶特価¥70,000
⑥GF-200（定価¥138,000）
▶特価¥89,000
⑦GF-240（定価¥148,000）
▶特価¥98,000

プリンター（送料¥1,000 消費税別）

（ケーブル 用紙 付）

■CZ-8PC5-BK
（定価¥96,800）
▶特価¥68,500
■CZ-8PK10
（定価¥97,800）
▶特価¥71,000

モデム

■PV-M24B5
（AIWA）（定価¥39,800）
▶特価¥25,000
（送料・消費税込み¥26,780）
■MD-24FB5V
（オムロン）（定価¥39,800）
▶特価¥25,500
（送料・消費税込み¥27,295）
■FMMD-311G
（富士通）（定価¥35,800）
▶特価¥24,800
（送料・消費税込み¥26,574）

P&A特選パソコンラック（消費税別）（送料無料）

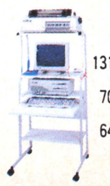
①3段¥7,900

1230(H)×600(D)×650(W)
（消費税込み¥8,137）

②4段¥8,800

1250(H)×700(D)×640(W)
（消費税込み¥9,064）

③5段¥12,500



1310(H)×700(D)×640(W)
（消費税込み¥12,875）

●全機種＝移動自由（キャスター付）●コードクランプ付（4段/5段）※5段のみ＝電源コード付（2.5m）（2P）キーボード収納可能

●本広告の掲載の商品の価格については、消費税は含まれておりません。 ●営業時間＝平日AM10:00〜PM7:00、日祭AM10:00〜PM6:00

アフターサービス万全
　全商品保証付。専門の担当がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

★頭金なし!!
★即日発送!!

秋葉原でおなじみのP&Aがズバリ超特価セールでご奉仕!!

● お近くの方は、お立寄下さい。専門係員が説明いたします。
● 本体単品でも受付します。詳しくは、お電話にてお問合せ下さい。
● ビジネスソフト定価の15%引きOK!! TEL下さい。
● 現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

全国通販

X68000用ソフトコーナー（送料1ヶ～5ヶまで￥500・消費税別）

◆Z's STAFF PRO68 Ver3.0(ツアイト)	定価￥58,000	▶特価￥37,500
◆Z's TRIPHONY デジタルクラフト(ツアイト)	定価￥39,800	▶特価￥27,000
◆テラッツォ(ハミングバード)	定価￥19,400	▶特価￥13,600
◆マジックパレット(ミュージカルプラン)	定価￥19,800	▶特価￥14,200
◆たーみのる2(SPS)	定価￥17,800	▶特価￥13,000
◆Mu-1 Super	定価￥39,800	▶特価￥28,500
◆CMA68K(シティソフト)	定価￥29,800	▶特価￥21,800
◆サイクロンEXPRESSα68	定価￥98,000	▶特価￥69,000
◆C-TRACE68 Ver3.0(キャスト)	定価￥98,000	▶特価￥68,500
◆G68K Ver.2 PRO	定価￥22,000	▶特価￥17,300
◆C&Professional Pack V3.2(マイクロウェアジャパン)	定価￥80,000	▶特価￥57,800
◆ウエットペイント1～3(ウエーブトレイン)〔各〕	定価￥15,000	▶特価￥11,500
◆マチエール(サンワード)	定価￥39,800	▶特価￥28,800
◆Windex PRO68(JEL)	定価￥28,000	▶特価￥20,500
◆CZ-213MSD MUSIC PRO68K	定価￥18,800	▶特価￥13,200
◆CZ-214MSD SOUND PRO68K	定価￥15,800	▶特価￥11,300
◆CZ-215MSD Sampling PRO68K	定価￥17,800	▶特価￥12,500
◆CZ-220BSD DATA PRO68K	定価￥58,000	▶特価￥40,000
◆CZ-224LSD The 福袋 Ver2.0	定価￥9,900	▶特価￥7,400
◆CZ-225BSD Multiword Ver1.1	定価￥32,000	▶特価￥23,000
◆CZ-243BSD CYBERNOTE PRO68K	定価￥19,800	▶特価￥15,000
◆CZ-247MSD MUSIC PRO68K〔MIDI〕	定価￥28,800	▶特価￥20,500
◆CZ-249GSD CANVAS PRO68K	定価￥29,800	▶特価￥22,000
◆CZ-251BSD Hyper word	定価￥39,800	▶特価￥29,400
◆CZ-253BSD CARD PRO68K Ver2.0	定価￥29,800	▶特価￥22,700
◆CZ-257CSD Communication PRO68K Ver2	定価￥19,800	▶特価￥15,300
◆CZ-258BSD Teleportion PRO68K	定価￥22,800	▶特価￥16,900
◆CZ-261MSD MUSIC studio PRO68K Ver2.0	定価￥28,800	▶特価￥21,200
◆CZ-263GWD Easypaint SX-68K	定価￥12,800	▶特価￥9,800
◆CZ-265HSD New PrintShop Ver2.0	定価￥20,000	▶特価￥15,400
◆CZ-266BSD PressConductor PRO68K	定価￥28,800	▶特価￥22,000
◆CZ-267BSD CHART PRO68K	定価￥38,000	▶特価￥29,800
◆CZ-284SSD OS-9/X68000 Ver2.4	定価￥35,800	▶特価￥25,600
◆CZ-285LSD C-Compiler PRO68K Ver2.1	定価￥44,800	▶特価￥32,500
◆CZ-286BSD BUSINESS PRO68K Popular	定価￥28,000	▶特価￥20,500
◆CZ-287SS SX-WINDOW Ver2.0	定価￥12,800	▶特価￥9,800

★ゲームソフト25%OFF!!（一部ソフト除く）

周辺機器コーナー（送料￥500・消費税別）

①CZ-8NS1	定価￥188,000	▶特価￥133,000
②CZ-6VT1	定価￥69,800	▶特価￥49,500
③CZ-6TU	定価￥33,100	▶特価￥23,900
④BF-68PRO	定価￥19,800	▶特価￥14,400
⑤CZ-8NM3	定価￥9,800	▶特価￥7,200
⑥CZ-8NT1	定価￥13,800	▶特価￥10,000
⑦CZ-6BE2A	定価￥59,800	▶特価￥42,800
⑧CZ-6BE2B	定価￥54,800	▶特価￥39,300
⑨CZ-6BE2D	定価￥54,800	▶特価￥39,300
⑩CZ-6BF1	定価￥49,800	▶特価￥35,800
⑪CZ-6BP1	定価￥79,800	▶特価￥57,000
⑫CZ-6BM1	定価￥26,800	▶特価￥19,300
⑬CZ-6EB1	定価￥88,000	▶特価￥63,000
⑭AN-S100	定価￥36,600	▶特価￥26,300
⑮CZ-6SD1	定価￥44,800	▶特価￥32,500
⑯CZ-6BN1	定価￥29,800	▶特価￥21,500
⑰CZ-6BV1	定価￥21,000	▶特価￥15,200
⑱CZ-6BC1	定価￥79,800	▶特価￥57,000
⑲CZ-6BG1	定価￥59,800	▶特価￥43,000
⑳CZ-6BU1	定価￥39,800	▶特価￥28,500
㉑CZ-6PV1	定価￥198,000	▶特価￥142,000
㉒CZ-6BS1	定価￥29,800	▶特価￥21,500
㉓CZ-8NJ2	定価￥23,800	▶特価￥17,500
㉔CZ-6BL2	定価￥298,000	▶特価￥214,000
㉕JX-100S	定価￥89,800	▶特価￥44,000
㉖JX-220X	定価￥168,000	▶特価￥121,000
㉗IO-735XB	定価￥248,000	▶特価￥152,000
㉘LC-10C1H	定価￥598,000	▶特価￥459,000
㉙CZ-6CS1(674C用)	定価￥12,000	▶特価￥8,900
㉚CZ-6CR1(RGBケーブル)	定価￥4,500	▶特価￥3,600
㉛CZ-6CT1(テレビ・コントロール)	定価￥5,500	▶特価￥4,400
㉜CZ-6BP2	定価￥45,800	▶特価￥33,300

中古・高価現金買取り 下取りOK!!

■まずはお電話下さい。
下取り専用 買取り電話 ▶ ☎03-3651-1884　FAX. 03-3651-0141
■下取り・買取りで、お急ぎの方は、直接当社に来店、または宅急便にてお送り下さい。

買取り価格…完動品・箱/マニュアル/付属品付の価格です。

● 下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。（差額は、P&A超低金利クレジットをご利用下さい。）
● 買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。
● 近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、￥1,000,000までお支払い致します。

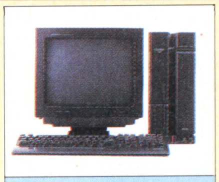
● 最新の在庫情報・価格はお電話にてお問い合せください。
● 買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せ下さい。
● 価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認下さい。
● 本商品の掲載の価格については、消費税は、含まれておりません。
● 現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

《便利な超低金利クレジットをご利用下さい》

● 月々￥1,000円からOK!!　● ボーナス払いOK（夏冬10回までOK）
● 支払い回数 1回～84回　● お支払いは、8ヶ月先からでもOK!!

● 定休日/毎週水曜日

P&A特選＝今月の中古特選品



● CZ-601C
● CZ-611D-TN
￥120,000

● CZ-634C-TN
● CZ-606D-TN
￥198,000

● CZ-644C-TN
● CZ-604D-TN
￥298,000

注目!!平成5年3月末一括払い手数料(金利)無料（平成4年12月末/平成5年1月末/2月末/3月末のいずれかをご指定下さい。）

買取り価格

● CZ-634C　￥150,000	● CZ-602C　￥68,000
● CZ-644C　￥200,000	● CZ-612C　￥78,000
● CZ-604C　￥80,000	● CZ-652C　￥48,000
● CZ-623C　￥110,000	● CZ-662C　￥68,000
● CZ-603C　￥78,000	● CZ-611C　￥58,000
● CZ-613C　￥90,000	● CZ-601C　￥45,000
● CZ-653C　￥68,000	● CZ-674C　￥150,000
● CZ-663C　￥75,000	

下取り交換差額表

新品 / 下取り	CZ-634C モニターセット	CZ-644C モニターセット	モデル UX20セット	モデル CX20セット	9801FA2
CZ-623C モニターセット	150,000	270,000	70,000	160,000	140,000
CZ-613C モニターセット	190,000	290,000	100,000	190,000	170,000
CZ-652C モニターセット	230,000	340,000	150,000	240,000	190,000
CZ-604C モニターセット	180,000	290,000	100,000	190,000	150,000
CZ-600C モニターセット	230,000	340,000	150,000	240,000	200,000

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
● 商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）
〔銀行振込でお申し込みの方〕
● 銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
（電信扱いでお振込み下さい。）

〔振込先〕さくら銀行　新小岩支店
当座預金　2408626 ㈱ピー・アンド・エー

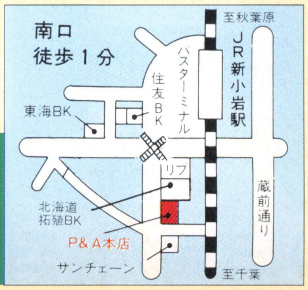
〔クレジットでお申し込みの方〕
● 電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
● 現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
● 1回～84回払いまで出来ます。但し、1回のお支払い額は￥1000円以上。

超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	3.0	4.0	5.5	5.5	8.5	11.5	16.0	21.0	27.0	33.0

マイコン専門ショップ P&A
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-3651-0148（代）FAX. 03-3651-0141

営業時間
平日：AM10:00～PM7:00
日祭：AM10:00～PM6:00



● 価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

MONTHLY SPECIAL
98+Macで万能ユーザーになる

すべての98ユーザーはMacを購入せよ！

●ダブルアーキテクチャのすすめ10のメリット ●セカンドマシンとしてのMacを知る（PowerBook／ClassicⅡ） ●いまや壁のない98とMacの互換性

●98とMacのパフォーマンスをテストレビュー ●98とMacのアーキテクチャはこう違う ●「常識です！」98&Macダブルユーザーの声

BEST BUY'92

- 386SL,HDD搭載の売れ筋ノートパソコンのベストは？
- 激戦区200Mバイトハードディスクの最高性能機はこれだ!?
- 若葉マークの人の統合ソフトのベストセレクトは!?

特別企画

486/66MHz DOS/Vマシン誌上体験
ハイエンドPC,486/66の驚くべきスピードを体感した！

PC SYSTEM UP

CONTURAとWord PerfectでバイリンガルPCに仕立てる
日米両用のDOS/Vと英語にも強いWord Perfect
5.1Jの組み合わせで国際派ユーザーに変身

別冊付録

実例トラブル解決法50例

SOFT BANK
ソフトバンク株式会社
出版事業部
東京都港区高輪2-19-13 NS高輪ビル
TEL.03-5488-1360

パーソナルコンピュータ総合情報誌

月刊PC

12月号/11月18日発売/毎月18日発売/定価650円（税込）

The スーパーファミコン

スーパーファミコンまるかじり！ 11/27号

定価380円(税込)
隔週金曜日発売

全国の書店、コンビニエンスストアにて発売！

巻頭！スクウェア2大特集 第2弾！

ファイナルファンタジーV オープニング全公開 潜水艦も登場!!

半熟英雄 シナリオ1 徹底攻略

青年期山場と仲間モンスターリストを紹介！

「ドラゴンクエストV～天空の花嫁～」

特集 RPG新時代へ向けて

PRGを創った男たちに聞く！ これからのRPGとは！

最新作をキャッチ・アップ★新作FRONT LINE▶タイニートゥーンアドベンチャーズ ▶SDガンダム外伝2～円卓の騎士▶スーパーテトリス2＋ボンブリス

読んで得するスーパーガイド★㊙新作SUPER GUIDE ▶ウィザードリィV▶餓狼伝説▶重装機兵ヴァルケン▶エルファリア

2大別冊付録
「SFCゲームカタログ」
「バルバロッサ」読本

BEEP! POWERFUL MEGA-MAGAZINE MEGADRIVE

ビープ！メガドライブ

12月号

好評発売中
定価480円(税込)
毎月8日発売

特集 ソニック・ザ・ヘッジホッグ2 超音速伝説ふたたび

第4回 メガドライブ・アカデミー賞前夜祭!!
全19部門！今年最高の作品ははたしてどのソフトに？

特報！G-SAT第1弾「ドラえもん」

新作スクランブル▶Jリーグ チャンピオン サッカー▶ベアナックルII

MEGA-CD PRESS▶3×3 EYES▶ぎゅわんぶらあ自己中心派2
▶電忍アレスタ▶アフターバーナーIII
▶カプコンのクイズの殿様の野望▶アネット再び▶ゆみみみっくす▶AYA

綴じ込み付録SPECIAL
ランドストーカー～皇帝の財宝～
地下潜入までの簡単便利GUIDE

SOFT BANK
ソフトバンク出版事業部

お近くの書店でお買い求めください

SOFT BANK

パソコン
はじめの一歩フェア

❖全国有名書店にて開催！

読書の秋、勉学の秋、パソコンの秋。
ご好評いただいているソフトバンクのブックフェア、今年は秋の登場。
初めての方も、ちょっとかじった方も、
深まる秋とともにもう一歩、奥まで学んでみては？
準備万全で、あなたをお待ちしています！

フェア取扱書籍　　定価は税込み

はじめの一歩　一太郎Ver.4	1,300円
はじめの一歩　Lotus1-2-3 R2.3J	1,300円
はじめの一歩　VZエディター	1,300円
はじめの一歩　花子Ver.2	1,300円
はじめの一歩　MS-DOS 3.3D	1,300円
はじめの一歩　エコロジーII	1,300円

新98NOTEスーパーブック	2,300円
IBM-PC AT互換機ガイドブック	2,400円
新MS-DOS入門　ビギナー編	1,900円
新MS-DOS入門　シニア編	2,300円
新MS-DOS入門　応用編	2,300円
新MS-DOS Ver.5.0入門　ビギナー編	1,900円
新C言語入門　スーパービギナー編	1,500円
新C言語入門　ビギナー編	1,900円
新C言語入門　シニア編	2,400円
新C言語入門　応用編	2,400円
まるごとFDブック	2,980円
C:98スーパーライブラリ	3,700円
Cプログラマのための アルゴリズムとデータ構造	2,200円
Practical C Programming	3,600円

その他にも小社の出版物を豊富に取り揃えております。
なお、現品が売れて補充中の場合もありますのでご注意ください。

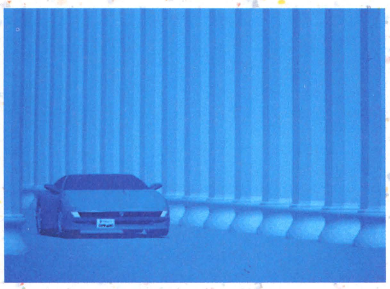
ソフトバンク出版事業部
〒108 東京都港区高輪2-19-13 NS高輪ビル TEL 03-5488-1360

11月20日
X68000版
新発売!!
新たなるロードス
「ロードス島戦記―灰色の魔女―」から数年がたち、
魔術師スレインは冒険者ギルドを組織していた。
しかし、黒騎士アシュラムと魔導士バグナードによって、
マーモ島からの黒い影がひそかに伸び始めていた。
ロードス島は再び戦乱の時代を迎えるのか?
そして、「五色の魔竜」とは?
ロードス島戦記の第2弾が
ついに完成しました。
快適な操作性を徹底的に追求、
MIDI対応サウンドと
グラフィックを
すべて一新して、
X68000ユーザーが
納得できる
傑作RPGに
仕上がりました。
■要メモリ2MB
■MIDI対応
標準価格
9,800円
原作:安田 均・水野 良 キャラクター原案:出渕 裕
ロードス島戦記Ⅱ©
―五色の魔竜―
※「ロードス島戦記Ⅱ―五色の魔竜―」は「ロードス島戦記―灰色の魔女―」をお持ちでなくとも遊べます。
ロードス島戦記
灰色の魔女
X68000版
好評発売中!
標準価格 9,800円
Humming Bird Soft
株式会社 エム・エー・シー ハミングバードソフト
〒530 大阪市北区曽根崎2丁目2番15号 TEL.06(315)8255
© Kadokawashoten/H. YASUDA & GroupeSNE
■標準価格に消費税は含まれておりません。お買い上げの際に別途消費税をお支払い下さい。
■通信販売ご希望の方は、住所・氏名・電話番号・商品名・機種名・メディアを明記の上、現金書留または郵便振替(大阪8-303340)にてお申し込み下さい。送料は無料ですが、標準価格に消費税の3%を加えた金額をお送り下さい。
No Copy
このマークは
不法コピー
禁止マークです

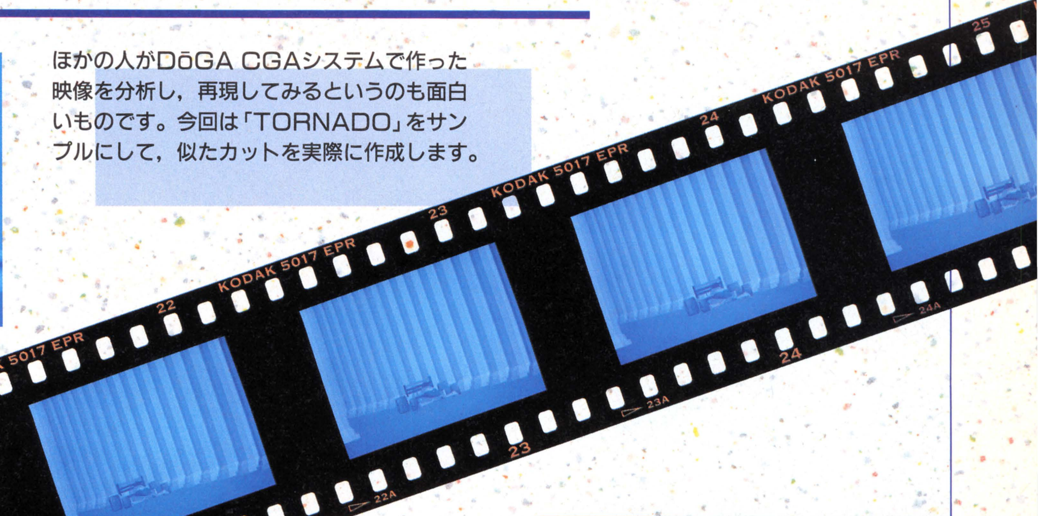
Oh!X Graphic Gallery

DōGA CGアニメーション講座

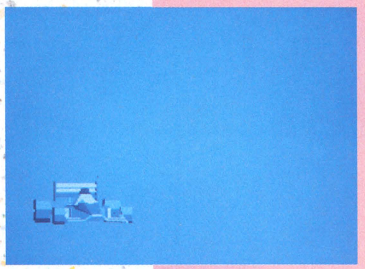
ほかの人がDōGA CGAシステムで作った映像を分析し，再現してみるというのも面白いものです。今回は「TORNADO」をサンプルにして，似たカットを実際に作成します。

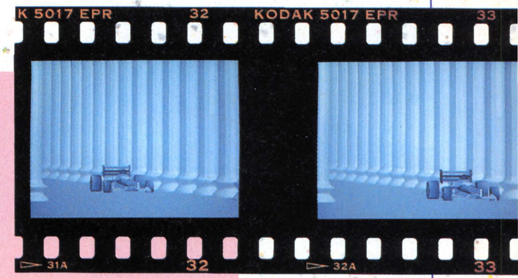


今回は「TORNADO」のこの1カットを真似してみます。

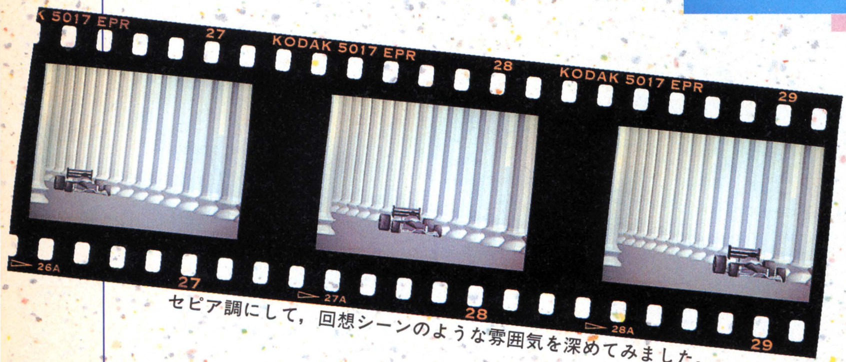


欧風の柱が並ぶ中を走るF1。青っぽい光源と空気遠近法による霞で画面全体が青みがかった色となります。また，画角を狭くして遠近感をなくしています。





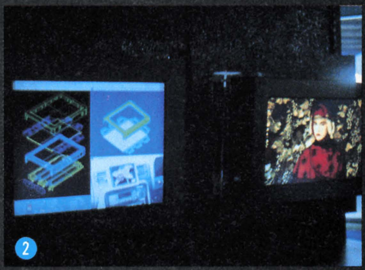
背景が動かない場合は，車だけのフレームに背景画像を合成します。
（NEGAで白黒に変換した画像）



セピア調にして，回想シーンのような雰囲気を深めてみました。

かまた氏と文月氏が共同制作した「XVIイメージデモ」の一部。車は旅のパーツであり，夕焼けのように寂しいものという思いが加わっています。

エレクトロニクスショウ'92



❶さすがにでかいフルカラーLEDディスプレイ
❷17型512色と16.5型1600万色の液晶ディスプレイ
❸ハイパー電子システム手帳用電子ブックドライブ
❹❺カシオはラジコンカーレースで液晶テレビをアピール
❻CDバトルもできるテレビ付きCDラジカセ
❼DCCもMDも試聴ブースが設置されていた
❽MDは素早い選局や携帯性を前面に押し出す
❾ハンディカメラはますます小さく，機能はアップ
❿ハイビジョンは簡易型から大画面まで幅広く出品
⓫コダックのPhotoCDではPhotshopでエフェクトをかけていた
⓬NEOGEO「龍虎の拳」でのチップのデモンストレーション
⓭CD-Iはもちろん数多く出品されていた

毎年，東京と大阪で交互に開催されるエレクトロニクスショウ。今年はインテックス大阪を会場に，10月13日から17日の5日間にわたって行われた。

エレクトロニクスショウでは，民生用エレクトロニクス機器や電子部品を中心とした展示がなされるが，今回はやはりデジタル録音再生機器のDCC（デジタルコンパクトカセット），MD（ミニディスク）が各社ブースで前面に押し出され，それぞれの特徴，ソフトの充実度，音質のよさがアピールされていた。

ブース別にいって面白かったのは，カシオとシャープであろうか（ひいきではない）。カシオはCDバトルができる液晶テレビ付きCDラジカセなどといった製品のユニークさもさることながら，アトラクションもラジコンのF1カーレースで盛り上がっていた。全体的に地味な雰囲気の漂うなか，元気さが目立つブースであった。

シャープはお家芸の液晶ディスプレイを軸に，液晶プロジェクター，液晶カラーテレビの新製品など，関連製品を多数展示。なかでも，ビデオカメラのファインダをそのまま4インチの液晶テレビに置き換えた，液晶ビューカムの試し撮りは人気を集めていた。コンピュータ用カラー液晶もより大画面になり，表示色数も1600万色と本格的なものが揃いつつあるようだ。

また，マルチメディアという言葉があちこちで使われているのも目についた。マルチメディア関連というのなら，展示物のすべてがそうではないかという気もするが，やはり時代の合言葉なのであろう。

ハイビジョンやCD-Iなども含め，すでに発表されている製品の延長に留まっているものが多く，目新しさがあまり感じられなかったのは残念だ。ただ成熟するというよりは，発展するものを見たいものである。

［第19回］
思索カード

寺尾　響子

響子inCGわ〜るど

それは，薄っぺらなプラスチックでできた，小さなトランプのようでした。

「さて，どれにしようかな。……ふうむ」

鼻の下にチョビヒゲを生やした医者は，手品師のごとくふんわりと扇型にカードを広げ，しばらく動きませんでした。

こんな症状はあまり見たことがないゆえに組み合わせのデータが存在しないゆえにああ困ったもんだ困ったもんだ。目がそういっていました。

「ええと，君の場合ですが。なにしろまれな例なんでね。処方カードを特注する必要があります。時間がかかるけど，いいですか？」

「はい」

「ああ，それから健康保険では扱えません。費用がどのくらいかかるか見積もりを出して，メールしておきましょう。IDは変更していませんね」

「そのままです」

「では，お大事に」

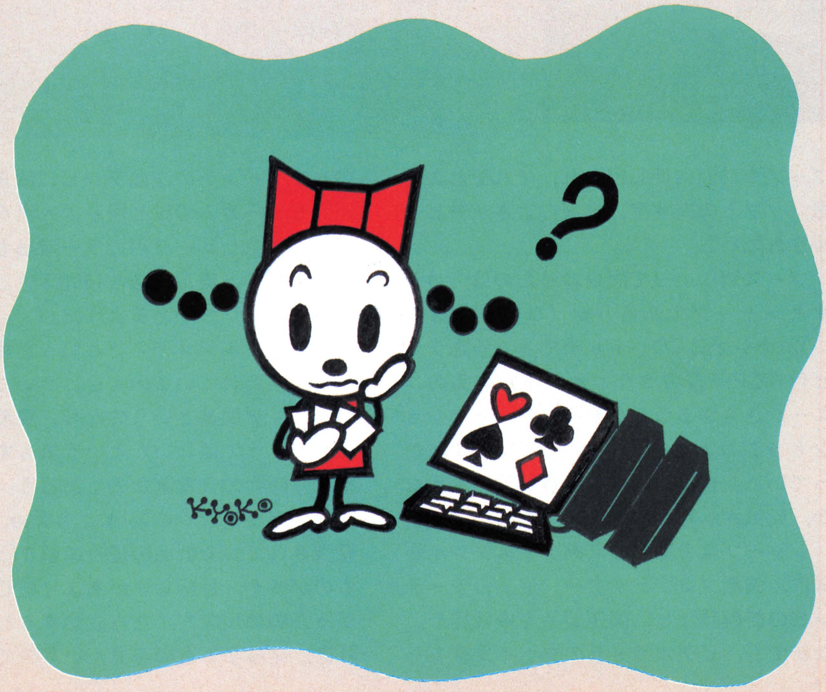
お礼をいう間もなく，ヒゲ医者はディスプレイの彼方に消えてゆきました。

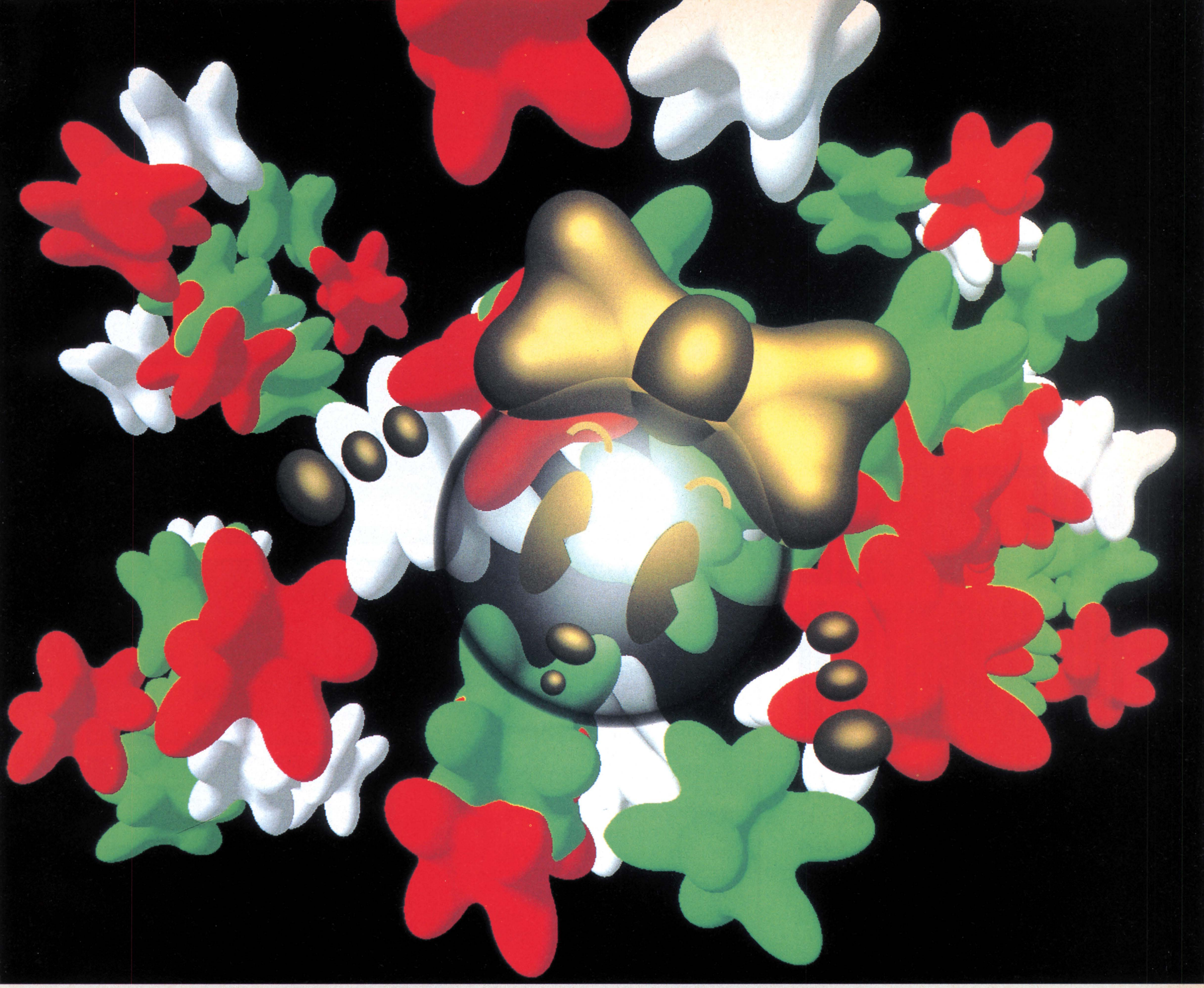
＊　　＊　　＊

同じような状況は幾度となくありました。受験のときは高い費用を払って，入試対策カード（もちろん学校別学部別です）を装着しました。学生時代は恋愛術カードがそれにとってかわりました。社会に出てからはスケジュール管理カード，人間関係保全カード。そして，いまの仕事に必要な文章構成力パワーアップカード（日本語版）が加わりました。

ここ何年かはそれで満足でした。メジャーな雑誌に連載を何本か抱え，単行本も数冊出しました。そこそこの売れっ子になっていたのです。たまにカードをバージョンアップして，時代の好みに合わせた文章を書けばよいというぐあいでした。

なにかが足りないのに気づいたのは最近です。生み出される文章がどれもこれも似たようなものばかりなのです。進歩のない再生産。工業ロボットのようにルーチンワークを繰り返すだけ。





自分の力で解決しようとはしましたがだめでした。お金を出してカードを装着するという安直な生活を長い間続けてきたために，脳が退化してしまったのです。手遅れでした。もう，自分自身の頭脳で処理ができなくなっていたのです。決まりきった思考にしか役立たない縮んだ脳みそ。

＊　＊　＊

予約時間になって，ディスプレイ上にヒゲ医者の顔が映りました。
「そちらにカードは届いていますね。君の頭には思索の領域が欠落あるいは不足していると判断しました。で，私自身が処方カードを設計しました。用意はよろしいですか？」
「はい」
「後頭部の空いているスロットにカードを差し込んでください。そうそう，カチッと音のするまでしっかりと。それから力バーを閉じてください。最後にリセットするのを忘れないで」

右耳の後ろにある小さいイボのような突起物を押しました。

暗転，そして目覚め。

いつもとたいしてかわりがないように思えました。いやむしろ，前よりもずっと頭の中がぼんやりとしていました。

あのヒゲ医者，ヤブ医者だったのかしら高い料金を払ったのに損したな有名な頭脳処理専門のエンジニアドクターだって聞いていたのにまったく。

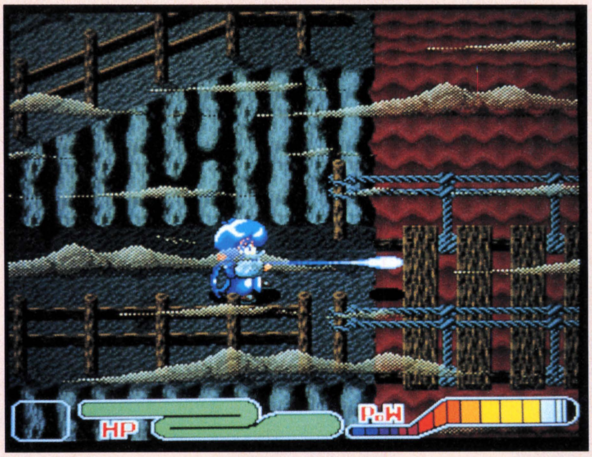
ふと，窓の外を見ると，雪が降っていました。湿った大きなぼたん雪がひらり，ひらりと散っています。今年ももうすぐおしまい。

振り返って頭の中を覗き込みました。とりとめのないもやもやとしたものが無数に漂っています。初めての感覚。

思索カードのせいに違いありませんでした。

SOFTWARE INFORMATION

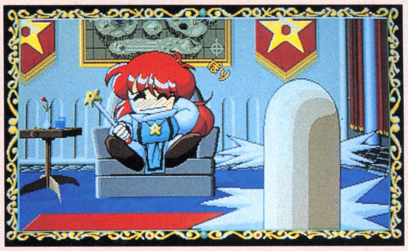
この本が出る少し前に「OVERTAKE」はすでに発売されているはずです。来月号で詳しく紹介するつもりなので，しばらくお待ちください。息を潜めていた「究極タイガー」も急浮上，近々発売されます。



エトワールプリンセス

今回は自キャラとして使えるメンバーとその特徴を紹介しよう。まず，主人公のリルル（星の部族)。通常攻撃は星屑が魔法の杖から発射される。特殊攻撃は流星が降るスターシャワー。セリナ（月）は遠隔操作が可能な杖を投げる。特殊攻撃は時計が降るスロー。ソル（太陽）は剣での接近攻撃。レーザーの特殊攻撃。ニース(地)は壁伝いに進むショット。そして，でかい足（画面一杯の）が敵を踏み潰してくれる特殊攻撃ゴッド・オブ・レッグ。ウリネ（水）は貫通ショット。特殊攻撃はホーミング。サーリア（火）はなにかにぶつかるか，一定距離で爆発するボムショット。特殊攻撃は火柱。シーリス（風）は攻撃力は弱いけど広範囲に広がるウィンドカッター。特殊攻撃は未定。

X68000用　5″2HD版　9,800円(税別)
エグザクト　☎025(247)9160



ライバルがいるから面白い

		(前回順位)
1.	OVERTAKE	2↑
2.	ファイナルファイト	1↓
3.	ストライダー飛竜	一初
4.	ポピュラスⅡ	3↓
5.	グラディウスⅡ	4↓
6.	ふしぎの海のナディア	10↑
7.	エトワールプリンセス	5↓
8.	シムアース	9↑
9.	出たな!!ツインビー	8↓
10.	ムーンクレスタ/テラクレスタ	一初

待ちに待った「OVERTAKE」が発売開始。アンケートハガキでこのゲームの名前を書いてくれた皆さんはもう遊んでいるでしょうか？

ここTOP10では，あれだけ強かった「ファイナルファイト」を発売前だというのに「OVERTAKE」し，見事にトップの座に輝きました。

これはあの「パロディウスだ！」以来の記録ですから，前人気のほうは十分以上のものがあります。X68000のレース物としては，いままでにないリアルさ，こだわりぶりですから，ロングヒットになると思われます。余談ですが，今年の日本GPはイマイチでしたな。

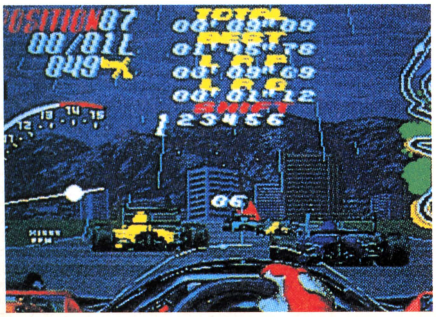
「ファイナルファイト」は首位から落っこちましたが，「OVERTAKE」が票を増したというよりは「ストライダー飛竜」と票を分け合ったところをやられたという見方ができます。ハガキのコメントの熱意はまだまだ衰えてません。

その「ストライダー飛竜」ですが，ゲームセンターで遊んだ人たちの心を確実につかんでいるのと，前作のおかげでX68000におけるカプコンブランドに対する安心感が出てきたことが，この初登場3位の要因として挙げられると思います。

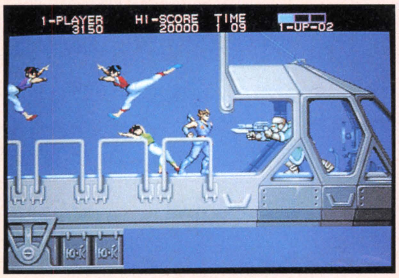
要するに「カプコンファン」というものが発生しつつあると。人体アクションのカッコよさ，破壊の爽快感みたいなものが，ユーザーを吸い寄せているように思うんですが。

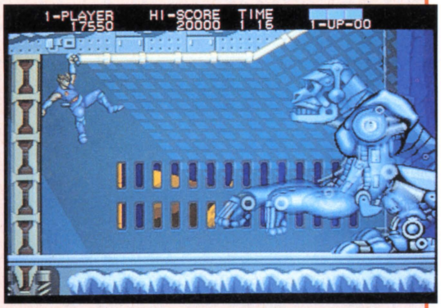
今月のニューカマーは電波新聞社の「ムーンクレスタ/テラクレスタ」。ちょっと古めの名作。主に20歳以上の人が「なつかしいし，いまでも遊べそう」と推薦してきています。

今月は「スターウォーズ」がランクアウト。いいソフトでも長くはいられないX68000のゲーム界。競争は激しいな。では来月まで。(浦)



ストライダー飛竜





ストライダー飛竜の武器はサイファー。刀型のプラズマ粒子を放出し，敵を叩っ斬る。まあ，そんな設定はともかく，剣で攻撃すると考えればいいだろう。そして，オプションアイテムを拾えば，３タイプのロボットが登場し，飛竜を助けてくれる。円盤型２足走行ロボット，ヒョウ型４足走行ロボット，鳥型飛行ロボット。後ろの２つだけを見るとバビル２世の気分だ！

Ｘ68000用　5″2HD版　9,800円（税別）
カプコン　☎03(3340)0718

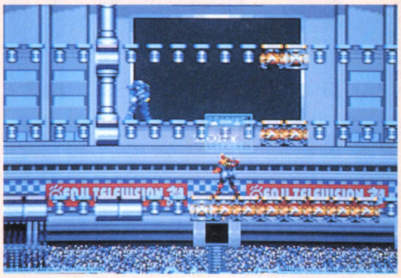
ストライクレンジ

ギミックハウスが「サンダーレスキュー」に引き続いて，第２弾を発売。今度はロボットどうしがバトルを繰り広げる対戦型シューティングアクションゲームだ。

多重高層になったドームで跳ね回るロボットたちを操り，相手の攻撃を避けつつ撃ち合わなければならない。画面はサイドビューで上下にスクロールする。

ロボットは「人型」「４足獣型」「３輪型」「キャタピラ型」「ドリル型」「液体型」（！）など，全部で12種類。

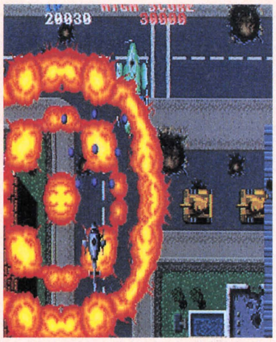
Ｘ68000用　3.5/5″2HD版　予価4,800円（税込）
ブラザー工業(TAKERU)　☎052(824)2493



究極タイガー

前作「飛翔鮫」発売から１年近くの月日が流れたが，ついに金子製作所Ｘ68000ゲーム第２弾「究極タイガー」が登場する。１面のみのサンプルをプレイしてみたところ，前作よりも移植度はかなりアップしていて，文句のつけようのない出来となりそうな気配。敵の動きもいいし，ボムも大きく広がる。これなら「究極タイガー」にはうるさい，という人にも満足いく仕上がりが期待できそうだ。発売は12月の予定。3.5インチ版はTAKERUで発売される。

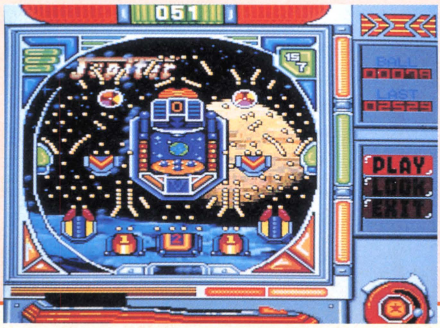
Ｘ68000用　3.5/5″2HD版2枚組　8,800円（税別）
金子製作所　☎0424(24)7752



パチンコワールド

謎の怪人に彼女をさらわれた。フィールドマップに点在するいくつかのパチンコ店で“打ち止め”しなければ，彼女を助け出すことができない……。「パチンコワールド」はこういう奇っ怪だがありがちなストーリーで開始するパチンコシミュレーションゲーム。もちろんパチンコ部分にもしっかり力は入っていて，全70台のパチンコ台で遊ぶことができる。

Ｘ68000用　3.5/5″2HD版2枚組　4,800円（税込）
ブラザー工業(TAKERU)　☎052(824)2493



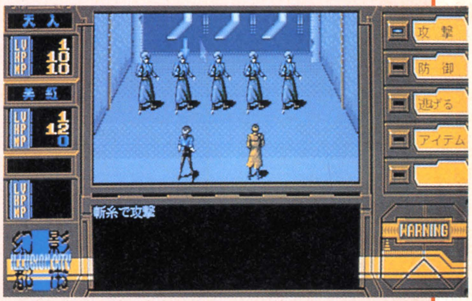
幻影都市

１年ほど前にPC-9801などで発売された，マイクロキャビンの伝奇RPG「幻影都市」がティールハイトによって移植され，TAKERUから発売される。

欲望と退廃，快楽と絶望が渦巻く幻影都市となった，近未来の香港を舞台に繰り広げられる秘密組織との戦い。数々の魔物も登場して，主人公の行く手を阻む。映画のような演出，VRシステム，８等身キャラクターが採用され，雰囲気づくりにひと役買っている。

Ｘ68000用　3.5/5″2HD版7枚組　6,800円（税込）
ブラザー工業(TAKERU)　☎052(824)2493

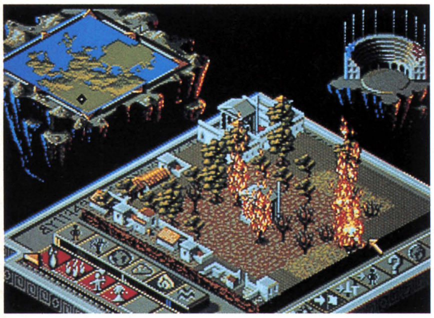
画面写真はPC-9801版のものです



TREND ANALYSIS

1992年9月の月間売り上げベスト10

POINT	タイトル	発売元	発売日
532	ポピュラスII	イマジニア	'92/8/28
192	バトルテック	ビクター音楽産業	'92/7/10
170	三國志III	光栄	'92/5/28
154	キャノンサイト	日コン連企画	'92/7/4
109	グラディウスII	コナミ	'92/2/7
107	アリスの館II	アリスソフト	'92/8/15
82	ネクタリス	システムソフト	'92/9/17
77	シュートレンジ	ビッツー	'92/7/24
93	ライジングサン	ビクター音楽産業	'92/8/28
54	太閤立志伝	光栄	'92/5/10



全体的に元気がないなかで,「ポピュラスII」がなんとかがんばっている。8月28日発売であるから,前回は4日間だけの売り上げ数だったことになり,それを考えると,勢いは前回のほうがあったことになる。しかし,スタートのダッシュよりも持続力が期待できる系統のジャンルであるから,順調に売れているといえるであろう。

ところで,この「ポピュラスII」の場合には,はっきりと評価の善し悪しに影響する要素というのがある。つまり,「ポピュラス」を気に入って「ポピュラスII」を買った人がいたとして(こういう人が大部分を占めるであろうが),その人は「ポピュラス」でどんな遊び方をしていたのか,ということである。極端にいうと,“対戦プレイ=「ポピュラス」の面白さ”であるという人かどうかということになる。

このゲームを面クリア型のパズルゲームと捉えている人は「ポピュラスII」に対して,かなり高い評価を下すだろう。使える業は多彩だし,見た目もずいぶん派手になっている。コンピュータが手強くなっているのも,前作からプレイしている上級者には腕の見せどころであろう。文句のつけどころがないといっても過言ではない。

しかし,“人間同士の対戦こそがすべて”と思っている人は,イマイチと感じてしまうのではないだろうか。上記のように1人プレイにおいてはかなりバージョンアップされているのだが,こと対戦に関しては特によくなったという点が見当たらないし,かえって複雑になった分だけ困ってしまうところが目立つ結果になってしまったかもしれない。

というわけで,前作からの“ポピュラス”シリーズの捉え方によって,評価が異なるであろうことは予想できる。しかし,その両者の比率のはっきりとしたデータがないし,この法則に必ずしも準じるとはいえないので,最終的な評価を読むのはたいへん難しい。それは,これから寄せられてくるアンケートハガキで,徐々にあきらかになっていくだろう。

初登場は6位の「アリスの館II」と7位の「ネクタリス」。「アリスの館II」はひさびさに登場のアダルトソフトである。このテのソフトはわりと人気もあるのだろうから,毎月なにか1本ぐらいベスト10入りしていても不思議ではないのだが,ここ数カ月は姿を見かけなかった。数が多すぎて票が割れていただけかもしれない。

システムソフトの「ネクタリス」はSFモノのシミュレーションゲームということで,「シュートレンジ」と同じタイプのゲームだが,元がPCエンジンだけにお手軽さという点では上を行っている。

10月はわりと多くの新作ソフトが発売されたが,下旬に固まっているようなので,どれほど食い込んでくるかはわからない。はたして,スタートダッシュのいいソフトが出てくるか。

[データ集計協力店](順不同)
九十九電機本店
J&P(渋谷/町田)
OAシステムプラザ横浜店
P&A
ラオックスGAME館

ウワサのソフトウェア（海外編）

Shadow of the Beast III

「ビーストIII出るらしいよ」。AMIGAユーザーは海外の新作情報を熱心に漁る。海外雑誌のレビューを見ていた秋川氏が教えてくれた。

「なんか変な帽子かぶってる」「結構遊べるみたいだよ」「うーん，いったい何があったんだろう」と好き勝手なことをいっているうちに，ようやく国内でも発売となった。

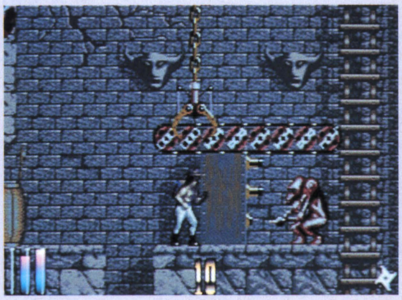
スクロールタイプのアクションゲームで，主人公を操り，立ち塞がる敵をぶっとばしつつゴールを目指すという，言葉で説明すればどうということのないゲーム。しかし，この「Shadow of the Beast」はシリーズを通して，ゲームそのものがもつ雰囲気が独特の世界を作り出していた。Iは純粋にアクションゲーム，IIは謎解きの要素が強い。両作品とも絵と音は最高の評価を受けたものの，ゲーム性についての評価はひどいものであった。少ない体力，厳しい敵の攻撃，きついハマリなどで，正攻法で制覇したものがいるかどうかすら疑わしい。

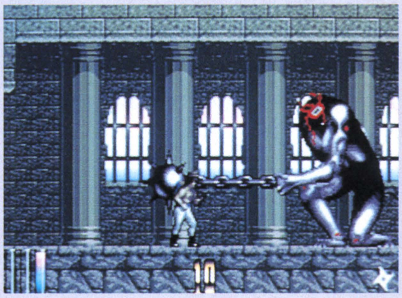
で，2年間の沈黙を破って発表された3作目。過去のシリーズを知るものすべてが驚いたに違いない。これがなんと，遊べる。画面の質感や音はそのままに，アクションはそれほど難しくなく，謎もハマリにくく，ハマってもすぐ回復できるように工夫されている。仕掛けはパズル的なもので，IIよりも解きがいのあるもの。演出も凝っていて楽しめる。

マニュアルを読むと制作チームのコメントがあった。IとIIに対するユーザーの評価を受け止め，それぞれのよいところを継承し，改めるべきところ（特にゲームバランス）は改めた，とある。細部まで凝りに凝り，技術的レベルもますます高くなっている。

この「Shadow of the Beast III」はこれまでのシリーズを良心的に進化させたものといえる。ただ，オープニングデモにはかつてほどのパワーのかけらも感じられなかった。　(A.T.)

発売元　PSYGNOSIS





ウワサのソフトウェア（海外編）

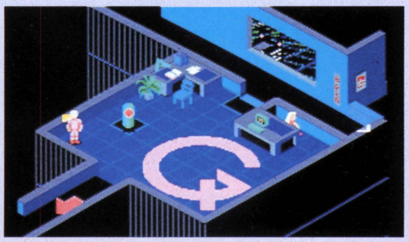
D/Generation

またハマりそうなゲームに出合った。基本的にパズル的なアクションゲームは好き。頭と指を適度に使うものがいい。ほどよい手応えのある暇潰し的なゲームには麻薬性がある。

主人公はバイオテクノロジーの研究所がある高層ビルにやってきた。そして，研究所の様子がおかしいことに気づく。バイオテクノロジーの生んだ奇妙な怪物が跳ね回り，生存者たちの生命は危険にさらされている。侵入者を抹殺するセキュリティシステムが作動していて，油断すると自分も殺されてしまう。

ビルの中の無数の部屋を回り，仕掛けられたトラップをかわしつつ，怪物を倒し，生存者を救助していくのがゲームの当面の目的。だが裏ではもっと大きな事件が進行していて，その全貌がしだいに明らかになっていくのである。

画面はそれほど派手なものではない。クォータービュー表示で登場人物がとことこと歩く。



いってしまえばそれだけのものである。画面も原色が多く，8ビット時代を思い起こさせる。

が，やっててなんだか楽しい。その原因のひとつはパズル的要素，もうひとつはプレイしてみた感触のよさである。

パズル的要素はゲームの舞台そのものである。さまざまな部屋のレイアウト，罠や怪物の巧妙な配置，これらがプレイヤーのヒラメキとジョイスティックさばきに挑戦してくる。佳境に入ると難しくなってくるのだが，やたら強い敵がどかどかと出てくるというのではなく，あくまで舞台の作り方で難しくしている。うまい。

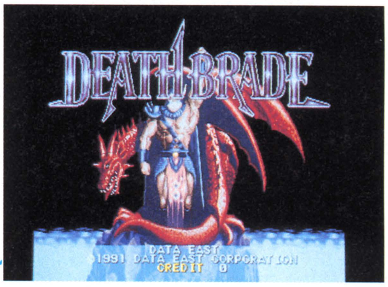
感触のよさはとにかく武器の気持ちよさである。光線銃は乱射でき，壁に反射する。跳弾を目標に当てる，手榴弾を投げて敵を爆破する。これらの動きが気持ちいい。音とグラフィックが実に小気味よく連動している。

比較的サクサクと先の面に進め，ハマりもほとんどない。それでいてやさしすぎるということがない。作りがしっかりしていて破綻がない。小粒ながらたいへん優れた作品である。

ひとつの大きな欠点としては，会話がうっとうしいということが挙げられる。大部分は無視するのだが，たまに必要な情報も含まれていてやっかい。ゲームの間に英語の長文を読みたくないというのもあるが，それ以前にゲーム進行のテンポが乱れるのがいやなのだ。　(A.T.)

発売元　Virgin

勝者に栄光を，敗者には闇を！



戦いとはきびしいもの。敵が強力な武器を持っていようと，火を吐く化け物であろうと，１対１で戦わねばならないときもある。「デスブレイド」はそんな硬派でアツい戦いをディスプレイに再現してくれる格闘ゲームだ。

Yaegaki Nachi
八重垣 那智

最近になっても対戦格闘ゲームの人気は衰えを知らず，盛り上がったままである。しかし，これがよく見ると基本的には相手から離れて戦うタイプばかりで，レスリングや柔道のように「組んで」戦うというタイプはほとんどないことに気づく。表向きはルール無用でありながら，パンチにキックと投げ，そして隠しコマンドのような必殺技というワンパターンの応酬では，作るほうも遊ぶほうも，いいかげん限界だと思うのだがどんなものだろうか？

猛者たちの宴

今回SPSから発売された「デスブレイド」は，1991年にゲームセンターにデビューしたデータイーストの対戦格闘ゲームからの移植作品である。特徴としては，平面をフィールドにしたプロレスっぽいバトルが繰り広げられる点と，登場するキャラクターがファンタジーの世界に出てくるモンスターを含んでいるため，独特の雰囲気を醸し出していることが挙げられるだろう。

私がこれを最初に見たときに真っ先に連想したゲームは，その２年前の1989年に同じデータイーストから発売されていた「ファイティングファンタジー」であった。これは奥行きのない舞台で，剣と魔法を駆使して魔物などと戦うというもので，まるでファンタジーRPGの戦闘シーンのみを抽出したという感じであった。

まあ，そのグラフィックや世界観が引き継がれているとはいっても，デスブレイドをその続編と呼ぶのは難しいかもしれない。しかし，データイーストは最近も，縦スクロールシューティングである「空牙」の続編的なストーリーで，横スクロールロボットアクション「ウルフファング」といった作品を作っているので，雰囲気のまったく異なる続編を作るのは好きなようだ。そう考えれば，このデスブレイドは立派な続編といえるだろう。

目指すは支配者の座

話がそれてしまったので元に戻そう。このデスブレイドの世界では，戦いによって優勝した者が国を支配することができるようになっている。つまりプレイヤーは，戦いに参加して勝ち抜き，現在の国王に挑戦，勝利して新たな国王になることが目標なのである。

ゲーム形式は２種類あり，優勝目指して戦っていく１人モードと，２人で争う対戦モードがある。自分の分身である戦士は８人の中から選ぶことができる。そのメンバーは，ファイター(人間)，アマゾネス(女戦士)，ヘラクレス(怪力男)，ワーウルフ(狼男)，ミノタウロス(牛男)，ビースト(怪人)，ゴーレム(鎧男)，ドラゴン(竜)といった面々である。

最初の３人は人間だが，あとは皆モンスターである。各キャラクターの能力は，パワーと防御とスピードの値の合計が同じくらいになるように設定されているようだ。つまり，「力の強い奴は遅い」といった感じで，特徴づけがなされているわけだ。

１人プレイ時に対戦する相手は全部で９人いる。１～３人目と４～６人目はプレイヤーが選んで，好きな順番で戦うことができる。その内訳は，セレクトで選ばなかったキャラクターから５人が登場し，続く３人(匹？)は特殊な能力を持つモンスター，そして最後の１人が現在の国王である魔法使いという構成になっている。

対戦相手の特徴はステージの前にアドバイスつきで表示されるので，見逃さないようにしたい。

力だけが正義

実際のゲーム中での操作はレバーで移動し，小技と大技の２つのボタンで攻撃を行うだけである。左右にはレバーを２度素早く入れるとダッシュでき，敵とある程度接近すると組むことができる。組んだ状態からはレバーとボタンの組み合わせで，投げやプレスなどの技を掛けることが可能だ。どんな技が出るかは，実際に試したほうがわかりやすいし，キャラクターごとに違うので，具体的には書かずにおこう。

X68000用　5"2HD版5枚組　9,800円(税別)
SPS　☎0245(45)5777

ドラゴンだから火を吐く。熱くて死ぬぜ

ドッペルゲンガーが自分に変身

そうやって戦っていると，パワーが溜まっていき，「OK」マークが点灯する。そうしたら組んだ状態からレバー上＋大技ボタンで必殺技を掛けることができ，大ダメージを与えることができる。このときのアニメーションはキャラクター特有でとても派手なので，自分で使えないキャラクターの必殺技に一度くらいワザとやられてみるのもいいだろう。これさえ決まれば不利な状況からの逆転も可能になるので，効果的に使いたいものである。

また，倒れている敵のそばで攻撃すると追い打ちができるし，ダッシュ中に攻撃することも可能。奥行きのあるフィールドをいかにうまく使うかがポイントになっている。面によっては，フィールドの中に触れただけでダメージを受ける物体もあるので，これもうまく利用したいところである。

試合は１本勝負で，勝利条件は敵の体力をゼロにすること。逆に自分の体力がゼロになるか時間切れでゲームオーバーになってしまう。もちろんコンティニューも可能で，その場合は減らした相手の体力が保持される。そのため，どんな相手でもいつかは倒せるようになっている。ここいらは親切設計になっているといえるだろう。

明日なき戦い

ゲーム自体の移植という面では，これはなかなかよい出来になっている。細かいグラフィックなども再現されているし，無用な解釈もないので，プレイの感覚にはあまりギャップを感じなかった。少し気にかかるとすれば，原作自体がそれほどヒットしなかった理由のひとつである，緻密さの少ない内容まで当然そのまま移植されてしまっていることである。

組み合って相手の掛けてくる技と，こちらの掛ける技の優劣を読み合い，互いに技を仕掛け合うという駆け引きが，格闘ゲームの本来の姿である。しかし実際のプレイでは，適当な技のボタンを連射しているだけの単調なプレイになってしまうのだ。これはつまり，組み合っているときの相手の動きが画面からでは若干わかりにくいというところに原因がある。

技を掛け合ったときの優劣があらかじめ示されていないことと，相手が何をしているのかよくわからないおかげで，どうして勝ち負けが決まったのかがはっきりしない状態に陥ってしまうのである。

このように，結果と自分の操作の因果関係がはっきりしないゲームは，ひどいものになると「クソゲー」などと呼ばれてしまうのだが，幸いこのゲームはそこまでメチャクチャというわけではない。納得できる範囲に収まっているので，これについてはあまり無茶はいわないでおこう。

しかし，そのあたりを納得したとしても移植そのものに，気になってしまうものがあったことは否定できない。つまらないことに文句をつけてしまって，ゲームの評価を下げてしまうことになるかもしれないが，やはりマズい部分は指摘しておこう。

邪魔者には制裁を

いちばん嫌になるのは，このゲームがフロッピー５枚組というシステムをとっていることである。つまり，頻繁にフロッピーの出し入れをしなくてはいけないのである。とりあえずハードディスクにインストール可能だが，それなりの容量を必要とするのが難点といえば難点である。たまにふと遊びたいときのために，それほど余裕のないハードディスクを割いておくのは，あまりいい気分のものではないからである。

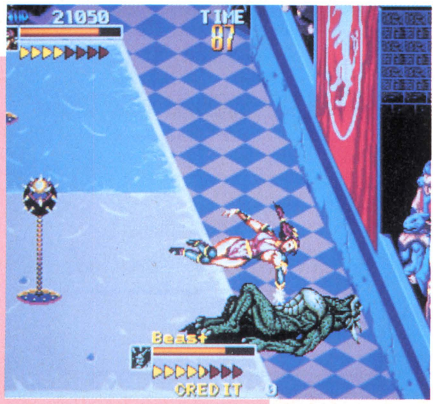
また，ハードディスクの使用を推奨するなら，インストール用のバッチファイル程度はつけてほしい。そんなに面倒な作業ではなかったが，やはりそういった気配りは重要だと思うのである。

メモリも２Mバイト必須とは書いてあるが，それ以上のオンメモリ動作に対応していない。やはりゲームは少しでも余計な作業を避け，のめり込むように遊びたいものである。そう考えると，ヘタすると毎面ディスク交換するようでは，テンションも下がってしまって楽しむのは難しい。たしかに慣れてくると，次にどのフロッピーを入れればいいのかはわかってくるので，イライラも多少は解決するかもしれないが，もう少し考えてほしいような気がする。

こういったシステム周りの問題を除けば，そこそこ遊べるいい作品になっている。オリジナルのファンでなくても，この力と技の世界に病みつきになってしまったら，もう二度とそこから逃げることはできない。デスブレイドは，遊ぶほどにハマる，怪しげな魅力を秘めたゲームなのである。

苦心の末の勝利。敗者は死神が始末する

お下劣な必殺技。わざとやられたりして



敵が倒れたらすかさず追い打ち。基本だ

勝者は常に孤独である

ゲームセンターにあった当時はさほどやらなかったけど，こうして落ち着いてプレイしてみると再発見もあったりして，熱くなってしまいました。しかし，キャラクターのセリフがかなりよく聞こえるので喜んでいたら，効果音で消されてしまうところもあり残念無念。こういった点も工夫してほしかったと思います。

ちなみにデータディスクの１枚に興味本位でLHAをかけてみたら，ほとんど小さくなりませんでした。圧縮してやっと５枚組なら，「なんちゃらII」なんか出たとしたらどうなっちゃうんでしょうねぇ。ちょっと心配だな，うんうん。

総合評価 (0–5–10)	
ゲーム性	★★★★★★★
技術	★★★★★★
サウンド	★★★★★★
グラフィック	★★★★★★★★
勝負の駆け引き	★★★★★
ヒドラの必殺技	★★★★★★★★★

愛と科学でナディアを救え

時は1889年パリ，人類は空に憧れ，その腕の中を自由に漂うことを夢見ていた。「ふしぎの海のナディア」はそんな時代に繰り広げられる海洋冒険譚だ。ジャンとナディア，そしてノーチラス号の運命やいかに。

Fuzuki Ryou
文月　涼

ジャン・ロック・ラルティーグは空に憧れる少年。彼の夢は自分で発明した飛行機に乗って，行方不明になった船乗りの父を探すことだ。

彼はパリの万国博覧会で不思議な少女ナディアに出会い，彼女が持つ謎の宝石“ブルーウォーター”とともに，次々と事件に巻き込まれていくことになる。

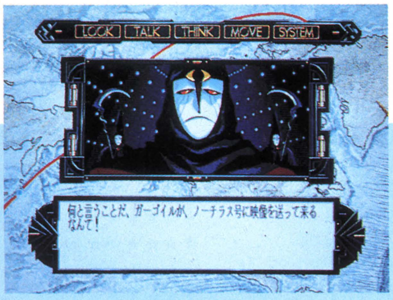
おりしも7つの海では，謎の怪物に各国の船が襲われ，次々と消息を絶つ事態が発生していた。だが，船を襲うソレは怪物などではなく，指導者ガーゴイルのもと世界征服をたくらむネオ・アトランティスの潜水艦だったのである。

ジャンとナディアは“ブルーウォーター”を狙うグランディスら3人組に追われることになるが，彼が発明した試作動力飛行機“エトワール・ド・ラ・セーヌ8世”で，からくも海へと飛び立つ。

こうして，一時は窮地を脱したものの，“ブルーウォーター”を狙うもうひとつの魔の手，ネオ・アトランティスにより，2人は飛行機を失うはめになる。

海に投げ出され，まさに息絶えようとしたそのとき，突然海中から浮上した潜水艦に救出される。それこそが悪の秘密結社ネオ・アトランティスにたった1艦で立ち向かう，ネモ船長のノーチラス号だったのである。

こうして冒険は始まる。



X68000用 3.5/5"2HD版9枚組 14,800円(税別)
ガイナックス　☎0422(22)1980

起動完了，本編に突入！

「誘導弾来ます」

「対雷撃防御，ホム・ガードを使え」

「了解。ホム・ガード発射！」

測的長の報告とほぼ同時に船長の指示が飛び，副長のよく通った声で迎撃が復唱される。瞬時にホム・ガードが発射され，誘導弾は爆音とともに艦の前方できらめく。

「弾幕突破！　誘導弾抜けてきます」

測的長の悲鳴にも似た声が，撃ちもらした魚雷の接近を伝える。

「全艦耐爆防御，衝撃に備えよ」

「全隔壁緊急閉鎖！」

船長と副長の指示が鋭く飛ぶ。一瞬右舷に誘導弾の走り抜ける音がし，次に爆音と強い衝撃が艦を襲う。しかし弾が爆発したほどの衝撃ではない。

「右舷被弾，不発です」

「敵艦の音紋解析の照合結果出ます。適合データは……ばかな！」

測的長は手にした分析結果を疑った。

「なんだ？」

「ノ，ノーチラス，本艦です……」

「なに？」

「あっ，敵艦失測！」

腕組みした船長の目が帽子の下で光る。

「ガーゴイルめ。とうとう我が艦と同型艦を発掘したのか」

そのことばに艦橋は静まり返った。

「たしかにそのとおりじゃ。なるほど反射光を後部噴出口から洩らしておるわい」

白い髭をたくわえた年輩者が噛みしめるようにいう。

「機関長，ということは」

「敵も不完全ながら対消滅機関を積んどるということじゃ」

機関長のことばは複雑な気持ちに追い討ちをかけた。

「追跡しますか？　船長」

操舵長が尋ねる。

「いや，右舷の不発弾を処理するのが先だ。どのみち，いずれ奴とはまた顔を合わせることになるだろう。第1種戦闘体制解除。各部損傷箇所の復旧にかかれ」

Here Comes Nautilus

はい。ファンの皆様には涙モノの「ふしぎの海のナディア」がアドベンチャーゲームになりました。ご存じのとおり，先にPC-9801版が発売され，好評を博していたのですが，私はひたすらX68000版が出るまで待っていたのです。その甲斐あって，仕上がりは上々。

アニメを知らなくてもマニュアルの冒頭に目を通せば，楽しめるストーリーになっています。

また本編のナディアのストーリーをあと1回だけでもいいから追いかけてみたい，と思っていた人には「ちょっとだけよ」の1回となるでしょう。

ゲームが始まるとネオ・アトランティス

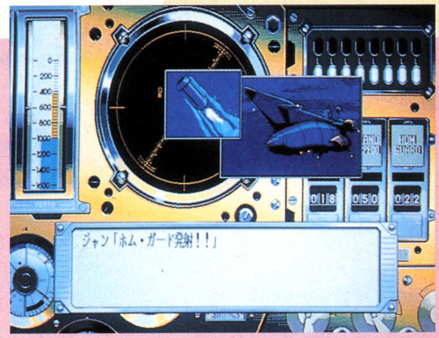
料理しつつも，どこか怪しいナディア

浮かび上がった姿はノーチラスだった

の指導者ガーゴイル様がテーマに登場します。そしてお馴染みの指パッチンと「これが君の最期となるのだ。ネモ君」のセリフです。舞台は変わってノーチラス号の艦橋。素早いテンポで，手に汗握る戦闘シーンが展開されます。これが前の章で紹介した場面なのです。

ちょうど艦橋に居合わせたところで戦闘状態になってしまい，外に出られなくなってしまうジャンとナディア。敵は手強くノーチラスも誘導弾（いわゆる魚雷）を食らってしまいます。誘導弾は右舷に着弾，しかし不発。結局敵は取り逃がしてしまうのですが，驚くべきことに敵艦の音紋はノーチラスのものと一致するのです。「ガーゴイルがノーチラスと同型艦を発掘した」という衝撃が艦全体に走ります。

危うしノーチラス号。運命やいかに



ホム・ガード発射，誘導弾を防げ

ナディアはいずこに

この戦闘状態から抜けると，やっと自由に動き回れるようになります。メインコマンドには「LOOK」「TALK」「THINK」「MOVE」「SYSTEM」の５種類が用意されていて，このそれぞれのコマンドを押すと，サブコマンドが表示され，それを選択して話をしたり行動したりします。ただし，メインコマンドは５種で固定なのですが，サブコマンドはその都度キーワードが追加されたり消えたりしますので，行き詰まった場合は「SYSTEM」を除いた残りの４つを，ひととおり見る必要があります。必要な知識はそれぐらいですので，お話を先に進めましょう。

戦闘状態が解けると同時に，ナディアは艦橋から行方不明になってしまいます。ジャンはナディアを捜し，格納庫，機関室，医務室，損傷箇所，マリーの部屋などなどを見て回ります。しかし，ナディアはマリーの部屋に現れたのを最後に消息がつかめません。

「おっかしいな。どこへ行ったんだろう」

頭をひねるジャンに，突然グランディスが駆け寄ってきます。

「ジャン，大変だよ。ちょっと来ておくれでないかい」

グランディスは有無をいわさずジャンをキッチンへとひっぱっていきます。そこで彼が見たものは，動物愛護主義者のナディアが魚を料理している姿だったのです。驚いた彼はグランディスと顔を見合わせてしまいます。

「どうしちまったんだろうねえ」

「あら，ジャン。おなかがすいたの？　もうすぐお食事よ。ちょっと待っててね。そうそう，先にこのお弁当をみんなに配ってきてくれない？」

目をうるうるさせてジャンに頼み事をするナディア。いやとはいえず弁当を配り歩くジャン。ナディアのお手製と聞いて我先にと群がるクルーたち。ジャンがお弁当を配り終わると，艦橋の面々が食堂に集まってきて，食事の準備も整っていました。めずらしくネモ船長も列席しています。

ナディアのお手製の夕食に，わき上がる歓声。なにか嫌な予感がして食事に手をつけていないジャンに対して「人間じゃない」ばりの非難が飛び交います。しまいにはナディアも泣き出す始末。追いつめられてしかたなく料理を口に運んだ瞬間，周りの人々が突然苦しみだし倒れます。食堂に響くナディアの高笑い。いったい何が起こったのでしょう。

「私の名前は生体兵器303号！」

「……せいたいへいき？」

そしてストーリーは，ノーチラスとガーゴイルの真の目的を追って進んでいくことになります。

クルーを失ったノーチラスは？

本物のナディアはどこに？

ガーゴイルの目的とは？

乞うご期待。

やっぱりナディアだ！

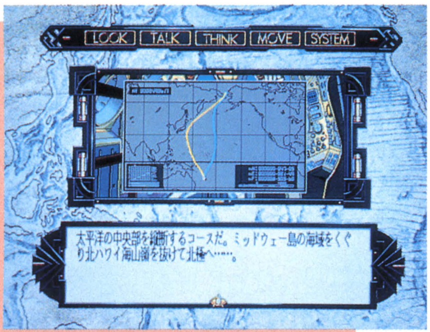
ストーリーはこのあと，複数の分岐点を経ながら進んでいきます。ハラハラドキドキ，混乱させられて「思わずゴメンナサイ！」の戦闘シーンもあり，飽きさせない展開と見せるテンポがマウスを動かす指を休ませないほどです。

はっきりいって，こいつは面白い。

長い間待たされたゆえに，かかる期待は過大に膨らんでいたんですが，その内容は期待を補って余りあるものに仕上がっているといえます。

画面にかぶりつくシーンこそありませんが，セーブしてから何度もトライしなければならない難関などもなく，ともかくストーリーはナディアフリークのみならず，一般の人々にも一度やってみていただきたい内容に仕上がっています。

ガイナックスさんはぜひ「ふしぎの海のナディア」をシリーズ化して，別のシナリオもバンバン出してほしいものです。値段は映画の入場料より高いけど，はるかに面白かったので。



こっちのコースはあれで，あっちは……

ナディアフリーク必買

本文でも触れましたが，あくまでもストーリーを追うものとして捉えた場合，高い完成度で仕上がっています。しかし，アクション性を求めたり，気合を入れてやろうとすると，人によっては肩すかしを食らうかもしれません。あくまでも，「ふしぎの海のナディア」の“メギドの雷”編なのであって，決してノーチラスを操る戦闘シミュレーションではないのです。

コマンドは複雑ではありませんが，同じような行動を選択する場合でも，別のコマンドの中に選択肢が出ることもあるので，進まなくなったら，各メインコマンド内のサブコマンドをよく探してみるといいでしょう。

総合評価	0　　5　　10
グラフィック	★★★★★★★★
サウンド	★★★★★★
ストーリー進展速度	★★★★★★★★
システムの操作盤	★★★★★★★★★
ジャンのエッチ！	★
ガーゴイルの押し出し	★★★★★★★★

仕事するならロードス島へ

Takahashi Tetushi
高橋 哲史

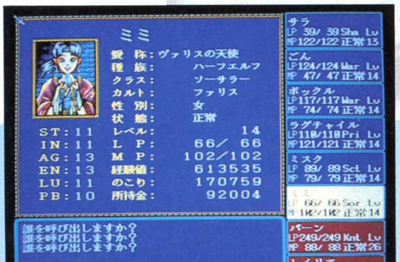
PC−9801版の発売からおよそ1年，やっとX68000でも「ロードス島戦記II」が遊べるようになった。初のMIDI対応などのオマケもついてなかなか楽しめる出来になっているようだ。頼まれた仕事を請け負って，最終目標へと突き進め。

ロードス島戦記はもともと「コンプティーク」でやっていた，“ロードス島戦記リプレイ”というテーブルトークRPGの実況中継（？）だったのですね。それをもとにゲームや小説，OAVなど（カセットブックやテーブルトーク版も出てましたっけ？）まさにクロスメディア的発展を遂げているシロモノなのです。しかも，ゲームデザインはあのグループSNE。中身の濃さは保証つきとゆーやつですね。ちなみにSNEってSyNtax Errorの略なんだそーで，初めて知ったときは笑ってしまいました。ま，だからどーだってことはないんですが……。

Record of Lodoss War

さて，暗雲渦巻くロードス島に冒険者たちは降り立ちます。“五色の魔竜”の最終目的はその名が示すとおり，五匹の魔竜たちの持つ古代祭器をすべて手に入れ，邪神ガーディスの復活を阻止することにあります。ぬおおと意気込む私はさっそくキャラクター作りを始めます（基本ですね）。

ロードス島戦記のキャラクターメイキングの特徴に，NPC（ノンプレイヤーキャラクター）の存在が挙げられます。あらかじめ性格や設定の決まったキャラクターが冒険者ギルドに待機していて，パーティに加えることができるのです。NPCはミッション遂行中に勝手にいろいろしゃべってくれたり，そのNPCの存在によってサブミッションの進行が変化したりするので，より深く演出されていくことになります。



X68000用　5"2HD版　9,800円（税別）
ハミングバードソフト　☎06(315)8255

なお，テーブルトークでいうところのNPC（物語の途中で順次加わっていく重要な役割をもつ仲間）はSPNPC（スペシャルNPC）として区別されています。写真を見るとわかりますが，赤く表示されているのがSPNPCです）。

しかし，私はとりあえずキャラを6人作り，NPCなしでパーティを編成しました。港街ライデンの市場に買い出しにいき，装備を揃え，いよいよ冒険に向かって準備万端となりました。ギルドに行って仕事の調達をします。ここではギルド長のスレインが温かく（？）迎えてくれます。

スレイン「サラさんたちは仕事をするのは初めてですね。まあ，最近南の洞窟で問題になっているゴブリンの退治でもしていただきましょうか」

がってんしょうちでいっ！　一行はライデンの街を出て，南にある洞窟へ。道中で発生する戦闘に戸惑いながらも（最初だから相手（モンスター）の強さも特性もよくわからないし，戦いにくいのだ）なんとかゴブリンを壊滅させ，またギルドに戻ります。しかしあのジョオウゴブリンのタフさにはまいったなあ。ちょっとグロいし……。

スレイン「ご苦労さまでした。報酬を受け取ってください。ところで先だって壊滅させたはずの盗賊ギルドなのですが，どうやら地下に潜って活動を続けているようです。加えて盗賊ギルド壊滅運動の先頭に立っていたフォースさんも，囚われの身になっているようなのです」

まかせといておくんなせぇ！　要は盗賊ギルドのアジトを突き止め壊滅させて，フォースさんを救出すればいいんですね。

与えられた情報をもとに，一行はどうも怪しいと噂されるライデンの酒場「紅い波亭」へ。酒場の地下がアジト（＝ダンジョン）になっていることを発見したパーティは，そこから無事フォースさんを救出して帰還します。このへんは練習問題みたいなミッションなのでサクサクいきましょう。

戦闘のほうもだいぶ経験を積んでスムーズに運ぶようになってきたと思います。だんだん楽しくなってきましたね。

このあとも隊商の護衛や，古代王国の廃虚探索などのミッションをこなしてレベルを上げていきます。このあたりから徐々に難易度も上がってきます。

と，ここで突然サブミッションがひとつはさまります。夜中に宿でぐっすり眠っているパーティにこっそりと近づく謎の影が……。

謎の男「……もし，ラグチャイルさん，ラグチャイルさん。起きてください。もし」

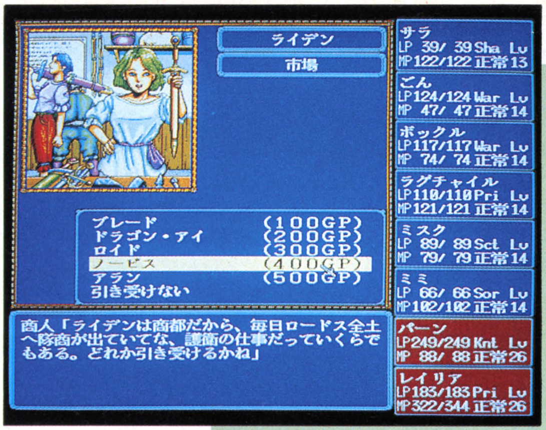
男は我がパーティの誇る有能なプリースト，ラグチャイル（通称：美しき彗星，恋人募集中）をそっと揺り起こします。

ラグチャイル「おや，あなたは宿のご主人。いったいこんな夜更けにどうしたというの

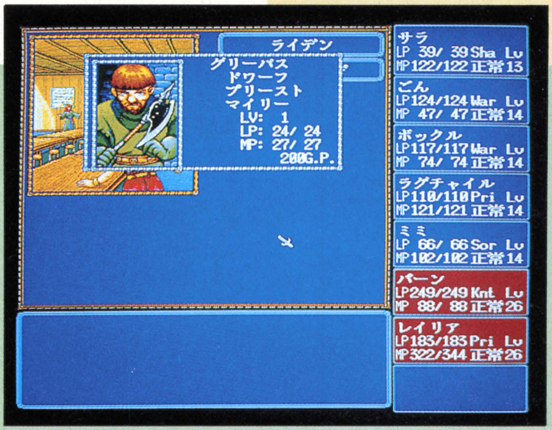
港から船が出ている。この先には……

デーモンとの戦闘。魔法を効果的に使え

です？」
宿屋の主人「貴方様を優秀なプリーストと見込んでお願いがあるのです。最近，毎晩のように地下室から妙な呻き声が聞こえるのです。客足にも影響が出始め，困っています。パーティの皆さんが揃って行動されるとまたなにかと噂になりますので，貴方様ひとりでなんとか原因を突き止めていただきたいのです」



ライデンでは隊商護衛の仕事もある



NPCのお方。ぼっちゃん刈りだ

ラグチャイル「ふーん，なるほど。呻き声がねえ。わかりました。さっそく地下室に行ってみましょう」

というわけでラグチャイルは単身地下室に乗り込んで，謎の呻き声の原因を探ることになります。宿に泊まっていきなりディスクがアクセスされたときは何事かと思いましたが，憎い演出ですねえ。思わずどきどきしちゃったじゃないですか。

さて数々のミッションをこなし，冒険者としての「ハク」もつくころには，いよいよミッションも本題に入ります。ここからは，あなた自身の手でプレイしてみてくださいね。

戦い終えて……

そんなこんなで合計３日ほどはまって，「ロードス島戦記II～五色の魔竜～」をクリアさせていただきました。いまの気持ちをひと言で申しますと，

「たいへん面白かったです。どうもごちそうさまでした」

といったところです。私は前作の「ロードス島戦記～灰色の魔女～」をプレイしていないので，前作との絡みがどうだとかはいえないのですが，その分IIだけをプレイしても十分楽しめる出来になっているということはわかりました。

全体的な流れとしては，本筋のミッションを順々にクリアしつつ，サブミッションも楽しんで，そこに適度な戦闘（経験値稼ぎ）がはさまるといったオーソドックスなRPGになっているのですが，それらのバランスがよく，またいわゆるハマリ（イベント飛ばしちゃってセーブしちゃったから元に戻れなーいとか，取るべきアイテムを取らなかったから先に進めないし，あとにも戻れなーい，なんて状態のことです）もないので，わりと「いつでも楽しい」という状態が続くのです。

宿のおねえちゃんはいつも明るい

ダンジョンの内部は不気味な暗闇

私もそれでついつい引き込まれてしまいました。さすがこのへんのデザインはこなれていてうまいものですね。特にサブミッションなんかはバラエティに富んでいて，いきなり殺人事件の捜査とか始まっちゃうし，わくわくしちゃいました。

あと戦闘時の戦略性もなかなか高いと感じました。武器によって攻撃もいろいろ変わるし，魔法もだてに数だけ多いのではなく，防御や攻撃，治療や召還などそれぞれがうまくバランスを保っていて，頭の使いがいのある戦闘が楽しめると思います。逆にいうと，タクティカルコンバットが本格的すぎて，毎回攻撃の指定をするのはちょっとうざったいともいえるかもしれませんが，シミュレーションに比べればずっと楽ですから大丈夫でしょう。それにターボ戦闘モードもありますから。

プログラムのほうもちゃんと手を抜かず移植してあって，画面は滑らかにスクロールするし，マウスオペレーションもしっかりしてるし，グラフィックもちゃんとX68000向けに修正が入ってるってことで好感がもてました。欲をいえば，ウィンドウの書き換えがちょっと遅いかなとか，戦闘が終わったあとノーマル画面に戻ってからちょっと間があるんじゃないかなとかあるんですが，まあ細かいことには目をつぶって満点に近い点数をあげちゃいましょう。

あと，実はMIDI対応なんです。MT-32，CM-64，SC-55，CM-500，TG-100と，出回っているあらかたの機種を網羅しているのもすごいんですが，さらに面白いことにRS-232C接続もできてしまうのです。もちろん変換ケーブルが必要ですが，MIDI端子にSC-55，RS端子にTG-100を接続して別々に鳴らすということをやってくれるのです。残念ながらうちにはケーブルがなかったので，実際に聴くことはできませんでしたが，音源をいっぱい持ってる方は試してみるのもよろしいのではないかと思います。たくさんの音源が同時に操作できるからって，必ずしも演奏がすごくなるとはかぎりませんが，やっぱりできると面白いですよね。

満足させていただきました

ああっと，読み返してみるとちょっとベタボメしてるみたいですね。でも，本当にひさしぶりに楽しませてもらったRPGなので，これが妥当な評価というところでしょう。わりと早めに終わってしまうあたりが，若干問題ありかなとも思いますが，新たなサブミッションを求めて2，3回は楽しめると思いますし，このくらいお手軽なほうがやっていて気持ちいいともいえます。お勧めの１本です。

総合評価	0　5　10
操作性	★★★★★★★★
シナリオ	★★★★★★★★★
グラフィック	★★★★★★★★★
音楽	★★★★★★★★

月と地球の熱い友情

Nishikawa Zenji
西川 善司

並の懐かしさではなく，ヒジョーに懐かしいゲームが電波新聞社から発売される。ニチブツの「ムーンクレスタ」と「テラクレスタ」だ。2本セットでお買い得な価格というのもうれしいけど，やっぱり“ドッキング”が目玉かな？

1980年代，日本物産株式会社というアーケードゲームメーカーは，なかなか侮れないゲームを連発していた。ハードやソフトも技術的にも優れていたし，それ以上にアイデアが奇抜だった。

ニチブツの代表作といえば今回，電波新聞社より発売される「ムーンクレスタ」と「テラクレスタ」のほかに「クレイジークライマー」がある。このゲームもアイデアが奇抜だった。なにしろビルをよじ登るという設定がすごい。さらにビルの住人が意味不明に鉢植えを落としてきたり，キングコングがビルに住みついていたり，「シラケ鳥」(死語)がうんこを落としてきたり……。舞台設定，グラフィック，サウンドと，当時のニチブツには，ほかにないキラリと光る独特のセンスとこだわりが感じられた。

そんなゲームメーカーの老舗，ニチブツの名作ゲームがX68000上に甦る。

ムーンクレスタ

世間でシューティングゲームの移動操作はレバーがいいか，ボタンがいいかという議論が交されていた頃のお話。

ガキだった私は，旅行先のホテルでろくに観光もせず1日中この「ムーンクレスタ」で遊びほうけていた記憶がある。旅先の記念品を買うために貯めておいた小遣いを全部注ぎ込んでしまい，ホテルのゲームコーナーの「ムーンクレスタ」では，ハイスコアのベスト1～5に私のイニシャルが毎日輝いていた。小遣いをすべて日本物産に巻き上げられた私は，ろくにお土産も買えずに帰路につくことになったわけだが，帰りの飛行場でも，未練がましく他人のプレイを覗き込んでたりしていて，親に「あんたって子は……」などと（当時は）理解不能なお叱りをうけたりもした。

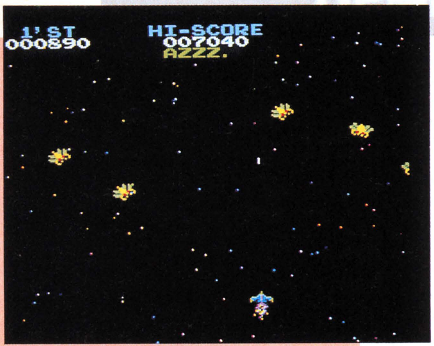
この「ムーンクレスタ」，BGMらしいBGMはなかったのに，なにか「音」というイメージが強く残っている。敵の登場とともに発せられるピロピロした音。なんとも緊張感をかりたてられた。敵によって音が変わるのと，敵の動きと同調して音が鳴るのがなんとも新鮮だった。また，周りがうるさくてもこのゲームの音だけはやたら大きく聞こえたのをよく覚えている。

また，自機登場のテーマや合体シーン，合体失敗のテーマはよくゲーマーの間で口ずさまれた。音楽が思いつかなかったのか，よく自作のゲームにもこれらのテーマがパクられていたのを見かけた。ムーンクレスタサウンドはまさにゲームミュージックのスタンダードだったのだ。

自機から発射されるショットの発射音と自機の爆発音がやけにリアルだったのも忘れられない。X68000版もほかの効果音はFM音源で鳴らしているのに，この2つだけはゲーム基板からのサンプリング音で演奏しているようだ。

また，「ムーンクレスタ」でもニチブツ特有の奇抜なアイデアは盛り込まれていた。持ち機が3機というゲームは当時でも珍しくなかったが，3機の機能が微妙に違うというのは斬新だった。1号機は小さく，単発ショットで弱々しい。1号機を失うと次に2号機の登場。2号機は2発同時に撃つことができ1号機よりも攻撃力が高い。3号機は少し機体が大きくなり2発撃てる弾の間隔が少し開いてしまうものの，なかなかに強力。そして，合体シーンで次なる自機と合体すればパワー倍増という仕掛け。

特定の機能しかもたない自機を操り，敵を撃ち落とすだけのシューティングゲームが世にはびこっていた1980年頃，私は「自機がパワーアップしちゃったりしていいの？」と感激のあまり目をうるませて沈みゆく夕日を両手を広げて仰いだものだった。当時の私に「パロディウス」やら「達人王」を見せたら，驚きのあまり鼻からそうめんを食べてしまったろう。



こうして見ると地味だが音はやっぱり派手だ

X68000用　5"2HD版　4,900円(税別)
電波新聞社　☎03(3445)6111

移植の出来

さすが電波マイコンソフトだけあって完璧な出来だが，本物は縦画面だった。普通なら無視される要因だが，さすがマニアの神様“電波”，縦画面モードも用意してくれてます。[HELP]キーを押すことで，ゲーム中でも切り替えられる。

タイトルデモから合体シーン，ネーミングシーンまでそっくりに再現されている。効果音もそっくりなので，私は初プレイのときには涙が止まらず男泣きしてしまった。キャラクターのケバケバしい色づかいや，背景のチカチカ流れる星もそっくりだ。変にゴージャスアレンジされていないので，部屋を片づけていて見つかった古い絵本をめくるときのような，懐かしさいっぱいのプレイができることだろう。

懐かしのドッキング，ピッピッピッピピピ

テラクレスタ

私は最初，「テレクラ・スター」というアダルトゲームかと思った。んなわきゃない。ぎょひょひょーん。

しかし，はずかしながら私は「テラクレスタ」をゲームセンターでプレイしたことがない。見たこともなかった。この「テラクレスタ」が発表された1985年という年は，ナムコ「バラデューク」，カプコン「エグゼドエグゼス」「戦場の狼」，コナミ「グラディウス」「ツインビー」などなど名作シューティングゲームが目白押しだった。私はもしかしたらそっちのほうに目を奪われて気がつかなかったのかもしれない。

しかし，いま挙げたタイトルを見て，現在あるシューティングゲームのルールの基本を作り上げたような名作揃いなのがおわかりいただけるだろうか。私は「テラクレスタ」をX68000上で初めてプレイしたことになるわけだが，このゲームに盛り込まれているアイデアの数々は，現在のシューティングゲームスタイルの形成に多大な影響を及ぼしていると確信した。1985年はまさにシューティング元年だったのだ。

ゲームシステム

自機「ウィンガー」の操作は8方向レバーで行い，ショットはAボタン，Bボタンはフォーメーションアタック（後述）のトリガとなる。敵には地上物と空中物がいるがどちらもAボタンから発射されるショットで破壊することができる。

時折，地上から自軍の秘密基地が出現し，これを撃てば自機と合体可能なオプションパーツが出現する。パーツを装着すると特殊武器が自機のショットと同期して使用可能となる。オプションパーツは5種類あり，全種類装着すると，一定時間無敵状態になる「火の鳥」に変身ができる。

また，なんらかのオプションパーツを装着すると，画面下部に[F]のマークが3つつくが，これは先ほど少しふれたフォーメーションアタックの回数を表している。この[F]マークの回数分だけウィンガーはフォーメーションアタックを仕掛けることができるのだ。フォーメーションアタックとは合体していたオプションパーツを切り放し，自機から遠隔攻撃ができるというものだ。フォーメーションアタック中のオプションパーツは無敵だ。新しいオプションパーツを取るたびに[F]マークは3つにリセットされるので，フォーメーションアタックは積極的に使うとよい。

パーツを手に入れれば武器の威力が増す

パーツを切り放せば特殊攻撃

オプションパーツはフォーメーションアタック以外に，身を挺してウィンガーを守ってくれる。自機が被弾したときにシールドの役目をしてくれるのだ（オプションパーツは失われる）。オプションパーツの捨て身の友情に私はしばし涙した。

敵の動きもとてもユニークでいやらしい。自機を巧みに画面端へ誘導し，ミスを誘う。7年前という過去を感じさせないアルゴリズムの巧みさもなかなか。このあたりのバランス設定のうまさも「名作（ANTHOROGY)」といわれるゆえんだろうか。

移植の出来

こちらも本物は縦画面だったというわけで，[HELP]キーでいつでも縦画面モードに変更ができる。ゲーム自体も速度的にも十分だし，1985年という時代を象徴するような色づかい，BGM，効果音がそつなく再現されているようだ。

さらに，国内ではBGMがPSGで奏でられていたそうだが，なんと海外版はFM音源(OPL)対応だったそうだ。そこで，毎度サービス精神旺盛の“電波”はこの両モードを再現してくれた。マニアならばムーンウォークでもして喜びそうだ。フゥー，マイケルあげたーい，いらなーい。

終わりに

古い名作ゲームはその姿を徐々に消しつつある。中古ゲーム基板市場でも，改めて買い求めるにはちょっと気兼ねしてしまうような値段がついているため，今回のような企画はゲームファンにはたまらない。

現在はゲームファンを引退中でも，あの頃はバリバリのゲーム小僧だった，という人も多いはず。そんな人はぜひあの感動をもう一度，自室のX68000で蘇らせていただきたい。

電波新聞社ではユーザーの反響次第で，今後も名作ビデオゲームをX68000へ移植，発売していくようだ。なにかリクエストがあればパッケージについてくるアンケートハガキにでも書いて送ってみよう。私はしつこく「ダイナマイトダックス」と「ラビオレプス」「ぶたさん」を書いて送るぞ。「ワンダーモモ」とかもいいな……。キリがないや。

ユーザーメイドを超えろ

当然のことながら，ゲームディスクにはプロテクトがかかっているので，作業の合間の息抜きにちょいと実行してプレイ，というわけにはいかない。いったんリセットしてゲームディスクを挿入しなければならない（いい忘れたが，ドライブ0から立ち上げると「テラクレスタ」，ドライブ1なら「ムーンクレスタ」となる）。

そのへんは残念といえば残念だがしようがないといえばしようがないか。しかし，ゲームの内容が軽ければ軽いほど（悪い意味じゃないよ）改めてフロッピーディスクから起動するということが「おっくう」になる。その点，最近盛んになっているユーザーメイドのゲームは，気軽に実行できるという点で軍配が上がる。内容的にもなかなかユニークなものが見受けられるし。最近は一度遊んでそのまま「ほったらかし」の市販ゲームソフトも多くなってきたので，うちではそういう，ハードディスクへインストールできるゲーム終了後は通常作業に戻れるというゲームが非常に重宝されているわけ。やっぱそういう条件を満たしているのって，ユーザーメイドゲームに多いんだよね。

総合評価	0　5　10
ムーンクレスタ	
アイデア	★★★★★★★★
ゲーム性	★★★★★★★
グラフィック	★★★★★★
サウンド	★★★★★★★
熱中度	★★★★★★★★
昔のよき思い出	★★★★★★★★★★
テラクレスタ	
アイデア	★★★★★★★★
ゲーム性	★★★★★★★★★
グラフィック	★★★★★★★★
サウンド	★★★★★★★★
熱中度	★★★★★★★★★
昔のよき思い出	★★★★★

アクションだって「ぐー」

グローディアの新作はなんとトップビュータイプのスクロールアクションゲームだ。バシバシ撃ちまくれるのはもちろんのこと，パートナーキャラも登場して戦ってくれる。なおかつビジュアルシーンはいつもどおり充実しているぞ。

Komura Satoshi
古村 聡

さて，今月の私の担当はグローディアから発売になった「バーンウェルト」です。グローディアのゲームをよりによって，また私にレビューやらせようってんですかい？　うーむ，今回は3歩さがって……。

物語の始まり

空間，時間，すべてが太陽系とは異なった世界がどこかに存在している。しかも，そんなところにも我々と同じような生命が生まれ，死んでいるのだ。

大洋上に突如，巨大船が出現した。

このニュースを聞いたトレジャーハンターのカノールはラーニアを連れ，巨大船に潜り込んだ。

そして，この宇宙船に興味をもった男がもうひとり。ゲーリー教授である。このマッドサイエンティストは失われた古代文明の力で世界を支配できると考えており，その古代文明を発見し，力を手に入れようという野望をもった男だった。そして，この男も自分のロボット，艦隊を引き連れて巨大船へと向かっていたのである。

巨大船の中でカノールとラーニアは船の動力室らしき部屋で，装置と共鳴してしゃべる金属板を手に入れた。その金属板は2人にこの巨大船が「ラーカイア」と呼ばれる宇宙船であること，船は「ヴェレアトール」と呼ばれる世界からやってきたこと，そして，そのヴェレアトールのエネルギー「ヴェルビアス」のことを伝えたのだ。

そこにゲーリー教授が到着。教授はカノールの持つ金属板を奪おうとした。カノールとラーニアが教授に追われ，ラーカイアを脱出しようとしたその瞬間，ラーカイアは突如，周囲の空間を巻き込み，再び時空転移を開始した。カノール，ラーニア，そしてゲーリー教授とその艦隊もともに転移し，次の瞬間に彼らは幻の古代時空都市ヴェレアトールにいた。

ラーカイアはどこへともなく飛び去っていった。ヴェレアトールに放置されたカノールとラーニアは元の世界に戻るため，ヴェルビアスを手に入れるための冒険がここに始まるのである。

初づくし

……という感じの今回のこの「バーンウェルト」なんですが，このゲームってなぜだか「初」づくしなんでありますね。

このゲームはめでたいことにグローディア初のX68000オリジナルゲームなのであります。グローディアといえば，いままでのゲームではたいていPC-8801がいちばん最初に出て，それからPC-9801版が発売されて，それからX68000にやってくる，というパターンだったように思うんですけど，今回はなんとX68000がいっちばん最初なのです。これはグローディアがX68000で本格的にやる気になったと考えていいんでしょうかねえ？　うむむむむ。

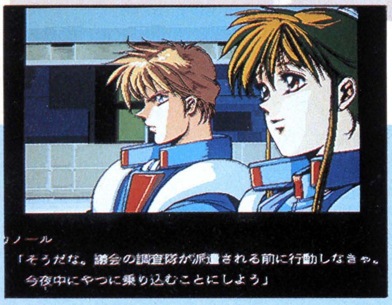
それと，たぶんこれもひょっとして初モノなんじゃないかと思うんですが(ちょっと自信はない)，この「バーンウェルト」はグローディアにしてはめずらしい，SF仕立てのゲームなのであります。グローディアといえば，いままで「サバッシュ」「エメラルドドラゴン」「ヴェインドリーム」「サバッシュII」と，ずーっとファンタジー系でした（「ヴェインドリーム」は，ちょっとだけSFっぽい部分もあるんだけど)。SFできる新人さんでも入ってきたんでしょーか？　むむむむむ。

さらに，この「バーンウェルト」はなんと，ロールプレイングゲームではなく弾撃ちバリバリのアクションゲーム（シューティングゲーム？）なのであります。画面写真を見ているだけだと，なんとなくロールプレイングかなあと思えてしまうんですけど（実際，弾を撃たないでマップの中をひたすら歩いているとロールプレイングゲームにしか見えない)，ゲームはしっかりとアクションシューティングゲームだったりするのです。むむむむむむ。

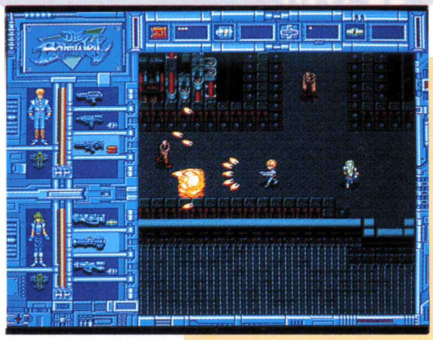
ついでにいうと私もシューティングゲームのレビューを書くのは初めてだったりするんですが。うーむ，ちゃんと最後までいけるのかなー。不安だ……。

舞台は廃墟から

ゲームの舞台はまず，カノールとラーニアの降り立った廃墟の街，フォレクスなの



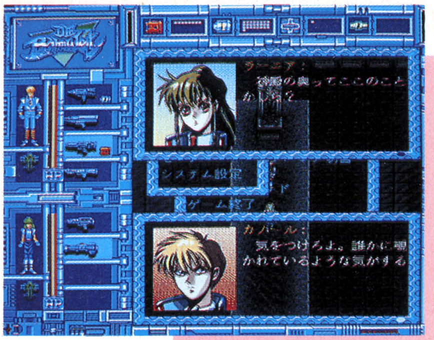
X68000用　5"2HD版2枚組　9,800円(税別)
グローディア　☎03(3220)5226



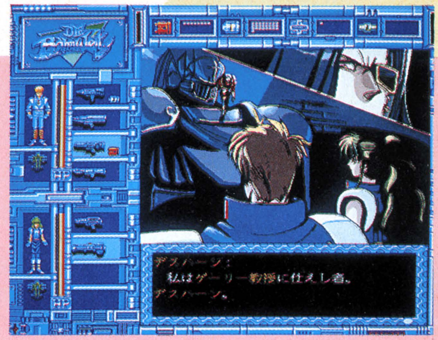
撃って撃って，どんどん突き進め

やっぱりあるのだビジュアルシーン

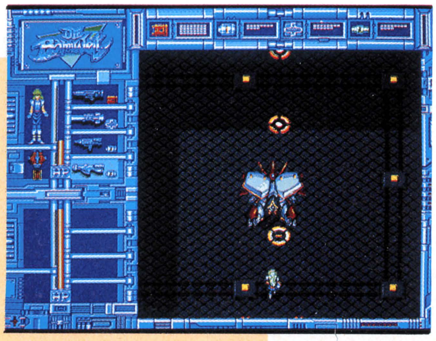
であります。
主人公はカノールを操るのですが，相棒のラーニアがほかのゲームでいうオプションのように（ちょっと動きは変わっているけど，AIというほどではない）ついてきてくれます。プレイヤーがどのキャラを操るか，あるいは誰が相棒なのかとかは，状況によって違うみたいなんですけどね。
で，この街は壊れかけた建物が林立し，地震によってできた地面の割れ目がそこかしこに横たわっているのであります。
街はトップビュータイプのフィールドで，アクションゲームというよりはロールプレイングゲームのような細い回廊なのですが，シューテングゲームですから弾はガンガン撃てます。
そうそう，このゲームでは複数の武器を持つことができ，そのなかから使う武器を選択できるようになっています。武器もさまざまなタイプがあって，エネルギーパックを消費してしまうもの，エネルギーパックは不要だけど威力の弱いものなど，いろいろあります。しかも，特定の武器を持っていないと倒せないボスとかもいて，武器という要素がけっこう重要な位置を占めるゲームであったりもします。
ま，なにはともあれ，そこでカノールとラーニアは出てきたメカどもをバッタバッタとなぎ倒しつつ，アイテムを拾いつつ，なにかありそうな，怪しげな神殿へとまっしぐらに進んでいくのであります。
神殿への道中には軍事基地があります。カノールとラーニアは軍事基地の中へと入っていくことになります。軍事基地を抜けないと神殿にたどりつけないんですね。
軍事基地の中には多くの兵器や都市のマザーコンピュータにつながる端末装置があり，ラーニアはそこから都市に関する情報，ヴェルビアスについて，そして都市のマザーコンピュータは神殿の内部にあることなどを知るのであります。
そして，次のステージは神殿。この神殿はかつては施政者の庁舎や議事堂といったものを兼ねていた，いわばこの都市の機構の中枢を一気に引き受けていた場所なんであります。



もちろん親切な相談コマンドもある



こいつが永遠のライバル，デスハーンだ


ボスキャラはめっちゃカタいぞ

神殿の内部には不気味な緑青色の像が並んでおり，こいつらは「この場より立ち去れ！」「さもなくば死あるのみ！」などと脅かしてくれますので，神殿内のどこかに落ちている信者の証なるものを手に入れて，さっさとフラグを立てなくてはなりません。
そして神殿の奥では，巨神像が並んでいる部屋に入るカノールとラーニアなのであります。カノールたちが巨神像に近づくと，あの金属板，レーデウスの鍵が共振を起こし始めます。何が起こったのかはわからぬが，巨神像はカノールたちにラーカイア，天空都市レクサー，レーデウスの鍵が揃ったときにヴェルビアスが復活することを告げるのであります。
しかし，これを陰から見ていた男がおります。その男はゲーリー教授の部下でサイボーグ，デスハーンなのであります。危うし，カノール！

かわいそ物語

ぜいぜい。結構飛ばして紹介するつもりだったんですが，物語の序盤をちらっとお見せしただけで終わってしまいましたね。
で，このあと物語はカノール，ラーニア，ゲーリー教授，デスハーン，それに何体かのロボット，アンドロイドなどの運命を濁流のように呑み込みながら，地底都市，海底都市，天空都市，そして天空都市レクサーへと進んでいくのであります。
で，結局，私はやっと天空都市までたどりついたのですが，ストーリーもそれなりに起伏があってなかなかであります。思わず「かわいそう」という感動的な章もありましたし。
そうそう最後になっちゃったけど，グローディアのゲームなのでビジュアルシーンもあります。
今回はいつものビジュアルシーン（ゲーム画面から急にアニメーションになって，映画館みたいになっちゃう，あれね）のほかにも静止画の子ビジュアル（ゲーム中のフィールド画面に静止画が表示される）まで入っていて，これまでにも増してビジュアル度満点です。
また，「エメラルドドラゴン」と同じく相談コマンドもありますので，それを使えば何をすればいいのか迷うなんてこともありません。
ま，この「バーンウェルト」，アクションゲーム＋いつもと変わらぬグローディアのノリってことで，結構よろしいんではないでしょーか？

伝統の味は変わらない

で，ゲームの感想なんですけど，どちらかというとアクションゲームの形をしたロールプレングゲーム，というかアクションロールプレイング全盛以前にあったゲームのような印象を受けます（西川善司氏は「テスタメントに雰囲気が似てる」といっていた）。主人公のヒットポイントがなくなってしまったときに，キャラがくるくる回って倒れるなんていうのは，なんだかあのころのゲームのノリを思い出しますもんね。いいか悪いかはともかく，最近のゲームってそういう部分がなくなってきたから……。
ということで，アクションゲームはめったに遊ばない私なのですが，個人的には気に入っています。あとはアンドロイドのエテルノがどうなるのか気になるので，なんとか最後までたどりつきたいなと思っているんですけど。
ま，ボスキャラがちとカタいのがアクション慣れしていない私にはつらいけど，なかなか不思議な味のするアクションゲームなんでありますよ，うん。

総合評価	0　　5　　10
マップ	★★★★★★★
ビジュアル	★★★★★★★★★
アクション性	★★★★★★★
音楽	★★★★
お買い得度	★★★★★★★

AFTER REVIEW

シミュレーション好きの人にはお待ちかね，多数の固定ファンを持つシリーズの第３弾「三國志III」です。「II」に比べて操作性が向上していることが評判よいのと，「複雑だけど長く遊べる」との声が多いようです。

三國志III

▶もう，いきつくとこまできちゃった感がありますね。初代で武将を引き抜いたあとヒヤヒヤしてたのがなつかしい。
亀田　徳隆(17)香川県

▶うーん，難しい。パラメータが多くておぼえられん。やっぱりマニュアル読まないと，ちょっと遊べないのね。武将の数がやたらと増えて，IIに比べて複雑になったけど，その分おもしろくなった。HEX戦でなくなったけど，HEXのときとたいして変わってない。「斬」のときもそうだった。でもおもしろければいいじゃん。
市川　徳明(18)東京都

▶操作のわずらわしさがなくなり，敵君主との緊迫感がいい。中野　義彦(34)新潟県

▶一騎討ちがおもしろい。呂布が曹操に負けるなんて，夢にも思わなかった……。とにかく，名作以外に言葉のない作品です。
桐本　順功(16)広島県

▶文官・武官で分けているので，結構，手間が省けてやりやすい。
花田　祐一(23)東京都

▶よーやくCD付きが手に入った。パッケージの青い文字が輝いて見えますよ。某国民機でも売れ線だ。　菅野　宗(22)茨城県

▶よいものはよいのだ。なんといっても，ちょーひ。　笠井　信幸(14)滋賀県

▶新君主８人。仮想武将うん十人。
越智　亮(20)大阪府

▶リアルになった戦い，外交を味わってほしい。　嶋崎　誠(18)千葉県

▶アイテムの要素がおもしろい。
木戸間　周平(14)神奈川県

▶難しくなった分，入れ込みがいがあってよい。　桜庭　宣裕(23)神奈川県

▶PC-9801版も持っているけど，X68000版は画面が広くなっていて，いい。
佐藤　剛(17)静岡県

▶マウスひとつで楽々操作……(IIだとマウスが使いづらかった……)。
石井　清貴(18)東京都

▶自分が作った武将が活躍してくれる。
本田　生二(30)滋賀県

▶オリジナルの武将が作れるのがよい。やはり女性武将にかぎる。私も最初に作ったキャラが春麗だった。ただし，君主じゃなくて武将だけど。しかし，名前３文字は少ないと思うぞ。最低５文字ぐらい欲しい。それと，カタカナとアルファベットは半角文字もあるほうがよい。でないと，60人も名前つけるのは大変だ。それから顔グラフィックの数も少ないし，美人も少ない。男はヒゲ面ばかりだし。内政や戦争訓練などに時間がかかるので，新君主だと非常にキツイと思う。10年以内にはムリかも？　しかし音楽のほうはウームだな。
岡田　伸一(24)京都府

▶操作性などがIIから比べ，格段に向上している。だてに３作も同じテーマで出していないと思った。　信太　徹(22)神奈川県

▶かなり理想的なゲームになってきている。あとは「天舞」のいいところを取り入れれば，最高になるかな。
岡崎　恭幸(19)大阪府

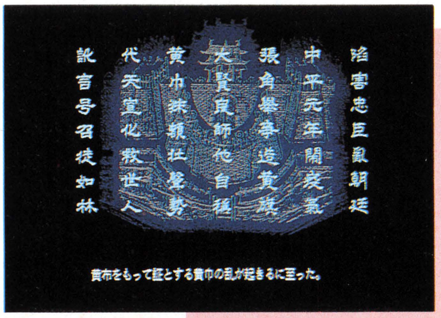
▶なかなか細かくできて，はまることができた。　橋本　久雄(16)香川県

▶おもしろい！　でも，ベタ移植の音楽はたいしてよくない。これだけX68000で光栄ユーザーがいるのだから，X68000用ソフトのミュージックをもっとよくしてほしい。　鈴木　雅之(19)東京都

▶シミュレーションの王様。
橋本　浩志(15)岡山県

▶システムが変わっていてよかったが，BGMが少々悲しい。もっと音数をフルに活用してほしいものだ。
小林　正弘(16)神奈川県

▶初代「三國志」をPC-8801でやってから，このゲームにはまっている。「三國志II」



はイマイチだが，今回は満足できる出来。はっと気づいてみると，いつも遊んでいるソフトである。アクション系のゲームはすぐに限界がくるが，このテのゲームは長く遊べる。　山本　昭治(24)神奈川県
▶ユーザーの希望に見事に応えてくれてます。　上田　洋史(20)神奈川県
▶とにかく三國志の世界が好き（待ちに待った）。　松本高佳(18)大阪府
▶純粋におもしろい。
遠藤　敬裕(26)広島県
▶おぉ～，メチャクチャ燃えるぜ～！
星野　弘孝(19)埼玉県
▶三國志は最高。　藤原　大介(18)富山県
▶16ビットらしいゲームだ。
中村　豊(26)奈良県
▶AとCの空き容量がそれぞれ0，1Kバイト。光栄とは思えない。
広瀬　良一(20)茨城県
▶ほぼ完璧。特に内政など……。
井場　秀樹(20)奈良県
▶グラフィックがいい。音はダサイ。
信垣　直嗣(19)大阪府
▶難しくてやりがいがある。
斉藤　大(17)埼玉県
▶前作より難しいけれど好き。
田沼　基司(26)茨城県
▶非常に永く(長く，ではない)遊べる。
益山　直人(18)北海道
▶やっぱりおもしろい。
野木　和洋(16)福島県
▶とにかくおもしろい。
丸山　藤市郎(16)滋賀県
▶よくできていて，おもしろい。
網干　学(21)宮城県
▶IIがかなりよい出来なので，IIIはそれ以上(?)になるかと思ったが……。でも結構ハマれる。　加藤　信之(22)東京都
▶はまる。　塩田　裕之(18)広島県
▶知らない武将がたくさん登場する。コンピュータから流れてくる曲はあまりよくないかもしれないけれど，CDのほうは一聴の価値あり。　古木　健一(18)神奈川県
▶結構いける。　新井　一成(22)埼玉県
▶難しくて，やりがいがある(でも，ちょっとダレる)。　宗京　邦和(20)奈良県
▶コンピュータが強い(HEXはアホだけど)。1国の中に90万以上の兵を持っているのがわかったときはビビった。
寺門　修司(21)兵庫県
▶やればわかる。　小林　正輝(15)北海道

▶僕は，いま出ているソフトでいちばん楽しいと思った。　波形　悦孝(18)神奈川県
▶めんどくさいけどおもしろい。でもやっぱりめんどくさい。もっとオートプレイの部分を増やすべきでしょう。
及川　剛(21)神奈川県
▶勉強になるかもしれない(うそ)。外交がよい。　海老原　公一(17)神奈川県
▶理由はなくとも三國志。やはり，光栄の戦国ものですから。青木　克央(21)岐阜県
▶IVを期待させるソフト。
古味　貴徳(18)千葉県

発売中のソフト

★サンダーレスキュー　ブラザー工業
X68000用　5″2HD版　4,800円(税込)
★ふしぎの海のナディア　ガイナックス
X68000用　3.5/5″2HD版　14,800円(税別)
★キャッスルズ　ビクター音楽産業
X68000用　5″2HD版　9,800円(税別)
★デスブレイド　SPS
X68000用　5″2HD版　9,800円(税別)
★バーンウェルト　グローディア
X68000用　3.5/5″2HD版　9,800円(税別)
★テラクレスタ＋ムーンクレスタ　電波新聞社
X68000用　5″2HD版　4,900円(税別)
★オーバーテイク　ズーム
X68000用　5″2HD版　9,800円(税別)

新作情報

★チェイスH.Q.　ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版2枚組　7,800円(税込)
★エアーマネージメント　光栄
X68000用　3.5/5″2HD版　11,800円(税別)
★エトワールプリンセス　エグザクト
X68000用　5″2HD版　9,800円(税別)
★ストライダー飛竜　カプコン
X68000用　5″2HD版　9,800円(税別)
★沈黙の艦隊　ジー・エー・エム
X68000用　3.5/5″2HD版　12,800円(税別)
★スクエア・リゾート　ハイパー戦車戦　ファミリーソフト
X68000用　5″2HD版　4,500円(税別)
★ファルディア　M.N.M Software
X68000用　5″2HD版　価格未定
★ロードス島戦記II　ハミングバードソフト
X68000用　5″2HD版　9,800円(税別)
★シムアント　イマジニア
X68000用　5″2HD版　価格未定
★餓狼伝説　ホームデータ
X68000用　5″2HD版　8,500円(税別)
★ヴェルスナーグ戦乱　ファミリーソフト
X68000用　3.5/5″2HD版　9,800円(税別)
★ドラゴンスレイヤー英雄伝説　SPS
X68000用　5″2HD版　価格未定
★究極タイガー　KANEKO
X68000用　5″2HD版　価格未定
★鮫！鮫！鮫！　KANEKO
X68000用　5″2HD版　価格未定
★達人　KANEKO
X68000用　5″2HD版　価格未定
★エアバスター　KANEKO
X68000用　5″2HD版　価格未定
★サバッシュII　ポプコムソフト/グローディア
X68000用　5″2HD版　価格未定
★極　ログ
X68000用　5″2HD版　12,800円

打倒TORNADOへの第一歩(完結編)

プロジェクトチームDōGA
かまた ゆたか

“打倒TORNADO”シリーズの完結編として，今回は，基本的なCGAシステムの使い方ではなく，さまざまな“表現”を行います。人の作品からテクニックを盗んで身につけましょう。

はじめに

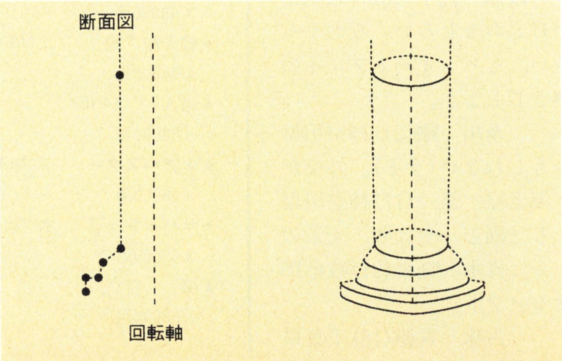
さて，4カ月かけてひとつの問題について論じていると，だんだん話の焦点がぼけてしまいますので，これまでの話を簡単にまとめましょう。

事の起こりは今年の4月，芸術祭において「TORNADO」という素晴らしい作品でグランプリを受賞した文月さんが授賞式で，「X68000とCGAシステムがあれば，このような作品は誰にでもできるんだ」と語りました。これは本当なのか，あるいはただの社交辞令なのか，というのがこの連載で取り上げている問題です。

あの作品をご覧になって，そのあまりの素晴らしさに，“とても誰にでもできるとは思えない”“少なくともCGの初心者である自分には100年かかってもできない”と思った方も多いでしょう。しかし，文月さんもCGどころかパソコンについても比較的初心者です。映像について特に勉強してきたわけでもありません（経済学部出身だったりして)。

ということで，とりあえず試しに車のCGを作ってみようということで，サンプルデータのF1を使って数カット作ってみました。すると，初心者にでも結構簡単にカッコいいカットができることがわかりました。また，初心者が抱えがちないくつかの問題点も，人の作品のマネから勉強するなどの方法で解決できそうです。とすれば，F1が走り回るようなカットを適当に作って，作品風にまとめるぐらい，どんな初心者にでもできるということです。打倒TORNADOの道程は，思ったより遠くはなさそうと感じたのではないでしょうか。

図1 回転体の柱



しかし，適当にカットをつなげただけでは，絶対に「TORNADO」に追いつくことはできません。CGAコンテスト（一般部門）に応募しても，入選することは難しいでしょう。それどころか，それは作品風になっているだけで，本当に“作品”と呼べるかどうかも疑問です。“作品”であるかどうかは“表現”の問題だと思います。今月はこの“表現”について考えてみましょう。なお，今回は少し上級者向けの内容となっておりますので，十分CGAシステムを使いこなしていない方にはわかりづらい点もあるでしょうが，あらかじめご了承ください。

TORNADOの1カットを盗む

当然のことながら，「TORNADO」は立派な“作品”であり，“表現”があります。この「TORNADO」を写真でしかご覧になったことのない方も多いでしょうし，前回，人の作品をマネる話をしたところですので，「TORNADO」の1カットを実際にマネて作ってみましょう。

マネるのは，Graphic Galleryに掲載しているカットです。欧風の柱が並ぶ中をゆっくりと近づいてきます。リアリティもあり，なかなか渋いですね。

このカットを制作するにあたっては，まず柱の形状を作らなくてはいけません。CADの基本的な使い方は，マニュアルの「CGA大学/CAD基礎実習（T-77)」ですでに習得しているものとして話を進めます（できれば「CAD応用実習（T-207)」の単位も取っておいてほしい)。初心者向けのCADの使い方に関する話は，また近いうちに行いますので，できない方も悲観しないでください。

○柱を作る

柱はCADの回転体の機能で作成します。まずCADを起動したら，「Attribute Mode」の「Attribute登録」でアトリビュート名を指定します。今回は1色しか使いませんから，「hasira」とでもしておいてください。「ESC」でメインメニューに戻って，「Edit Mode」に入ったら，画面右下隅の「atr no」をクリックして，アトリビュート

を「hasira」にするのを忘れないでください。
「回転体作成」に入ったら，角数「12」が表示されます。左右クリックすることで角数を変更することもできますが，今回はこのまま12角形でよいでしょう。「回転体作成」に入ったら，まず中心軸を指定します。今回はZ軸がそのまま回転軸になりますので，Z軸上の任意の2点，たとえば(0,0,0)と(0,0,400)をスペースで指定します。すると中心軸が黄色く表示されます。

次に断面図の形を作ります。半径150程度，高さ1800程度であれば，形など細かい点はなんでも結構ですが，参考までにひとつ具体的な数値を示すと，正面図で断面を描くとして，下から順に，

(0,-170,0)
(0,-170,40)
(0,-150,40)
(0,-140,80)
(0,-120,110)
(0,-120,1800)

の各座標に十字カーソルを合わせスペースを押します。最後の頂点(0,-90,1800)でスペースを押したあと，リターンで回転体が生成されますが，この最後の頂点は正面図の作画範囲よりかなりはみ出したところになりますので，画面上で座標を指定するのではなく，右下のZ座標値が表示されているところをクリックし，数値入力状態にしてから，キーボードで「1800」，リターンとするほうが早いでしょう。

柱が完成したら「ESC」でメインメニューに戻り，「File Mode」の「Save」で出力ファイル名を「hasira」としてセーブします。

柱以外の形状データとしては，地面があります。これは，XY平面の1枚の面で十分でしょう。大きさは，モーションデザインの段階で拡大できますので，適当で結構です。たとえば，

(400,400,0)
(-400,400,0)
(-400,-400,0)
(400,-400,0)

としておいてください。アトリビュート名は「ji」とでもしておきましょう。

○分析する

「TORNADO」のカットをマネするにあたって，いちばん簡単なのは文月さんにデータをもらう方法ですが，それでは意味がありませんから，映像だけを見て分析してみましょう。

写真からはわからないのですが，このカットを見て最初に気がつくのは，車がこっちへやってきてもほとんど大きくならない，つまり遠近感がほとんどないということです。望遠レンズを使って写真を撮ると，このように画面全体が妙に平面的になりますので，このカットでは，極端に画角を狭くしているのがわかります。

また，画面が青っぽく統一されており，まとまりがよいうえにリアリティが出ています。単に青っぽく統一するのなら，柱や地面などをすべて青っぽい色にするという方法もありますが，このリアリティは出せません。これは光源に色をつけているのだと思います。光源にちょっと色をつけることでリアリティを出すというテクニックは，応用範囲の広いものです。

さらに，画面全体が柔らかいというか，少し霞がかかったような雰囲気があります。霞がかかるといえば空気遠近法(マニュアルT-316参照)ですが，もし空気遠近法を使っているのなら，手前の柱と奥の柱では色が異なってくるはずです。しかし，特に色が異なっているようには見えません。これはちょっと悩みましたが，やっぱりこの柔らかさは空気遠近法しかありえないと思います。以上のように，このカットのポイントは，望遠，有色光源，空気遠近法の3点です。

また，表現のテクニックとは違いますが，このカットはフィクス（カメラが固定）ですので，背景が動きません。各フレームごとに同じ背景を計算するのは無駄なので，最初に背景を1フレームだけ作画し，あとで車を作画する際に，その背景画像を合成したと思われます。こうした工夫で，作画時間が大きく削減されます。

○アトリビュートを設定する

アトリビュートファイルが色や材質感を設定するファイルであることはご存じでしょう。これをデザインするのがATRです。コマンドラインから「ATR」で起動します。設定するアトリビュートは，「hasira」と「ji」です。この2つのデータを「back.fsc」というひとつのアトリビュートファイルに収めることにしましょう。

まずアトリビュート名を設定します。「未使用」と表示してあるところをクリックすると，小さいウィンドウが開いて入力状態になりますので，「hasira」リターンとします。同様に隣に「ji」を設定してください。

その下に並んでいる各パラメータは，マウスでインジケータを左右に動かすことで設定します。前のバージョンのATRでは，各パラメータの値が表示されていたのに，現在のバージョンではなぜかなくて不便なのですが，だいたい以下の値になるように，インジケータを調節してください（各インジケータは，0～1で，1目盛りが0.05)。

atr hasira

赤成分	0.9	|||||||||||||||||||
緑成分	0.9	|||||||||||||||||||
青成分	0.9	|||||||||||||||||||
透明度	0.0	
環境光	0.25	|||||
直接光	0.80	||||||||||||||||
反射光	0.3	||||||

散乱係数　0.1　||

色相　0.0

atr ji

赤成分　0.2　||||

緑成分　0.2　||||

青成分　0.4　||||||||

透明度　0.0

環境光　1.0　||||||||||||||||||||

直接光　0.0

反射光　0.0

散乱係数　0.0

色相　0.0

各パラメータの詳しい意味についてはマニュアルをご覧になっていただくとして，まず「hasira」のほうは，材質は石なので，反射光（スペキュラー）をおさえぎみにしています。写真では青みがかった色をしていますが，これは空気遠近法による霞の色であって，柱自身は白にしておきました。「ji」のほうの特徴は，環境光（アンビエント）が1で，ほかの材質に関するパラメータがない点です。これは光源の色や光線と面との角度などにとらわれずに，ズバリ設定した色になってほしいからです。ただし，今回は空気遠近法を使いますので，設定したとおりの色にはならず，空気遠近法による霞の色の影響を受けます。

各パラメータの設定ができたら，画面下中央のメニュー（図2）の「ファイル名」をクリックしてファイル名を設定します。「back」と入力してください。そして，「データ書出」でファイルをセーブしたあと，「終わりたい」で終了します。

○背景を作る

FFEで地面と柱を並べます。先にも述べたように，車「TYRL.SUF」はあとから合成します。

とりあえず，柱を並べてみましょう。「物体設定」の「追加」で「hasira.suf」を選択します。位置はX軸上6000～－15000まで1000間隔で，

(6000,0,0)

(5000,0,0)

⋮

(-9000,0,0)

というようにして，16本植えていきます。かなり範囲が広いですから，画面のスケールを変更するか，数値を直接入力する必要があるでしょう。

この作業はかなり面倒ですから，エディタが使えて，フレームソースの文法をある程度理解している方は，エディタで直接書いたほうが早いでしょう。その場合，まずFFEで柱を１本だけ立てて，そのファイルをセーブし，修正を加えるのが効率的です。

```
{ mov ( 6000 0 0 ) obj hasira (: HASIRA.SUF :)
}
```

という行をコピーし，movのX座標を書き換えます。このとき注意しなければいけないのは，そのファイルの4行目の「Objectnumber」です。この値はこのフレームソースファイル中にいくつの物体があるかというデータで，コメント文とはいえ，FFEには不可欠な情報です。柱をつけ加えたら，その数に応じてこの値を増やさないと，再びFFEで呼び込めなくなってしまいます。

柱はこの１列以外にもう１本，

(5000,-1800,0)

の位置に置きます。これは画面左手前の柱に相当します。柱を並べたら次に視点を設定します。「視点設定」で，

視点　(29000,-4500,500)

注目点　(-140,-200,500)

とし，さらに画角を「４」とします。通常の画角が60度なのですから，極端に望遠になっているのがおわかりいただけると思います。「決定」すると完成予想図が表示されますので，「TORNADO」の構図とだいたい同じになったことを確認してください。

次に忘れずに地面も置きましょう。とはいっても，よく考えると，この地面は省略可能です。「背景設定」の「べたぬり」でこの地面の色を設定すれば，柱以外のところはすべてこの色になります。しかし，今回は「空気遠近法」を使うため，「べたぬり」が使えません。「空気遠近法」と「べたぬり」の併用ができないのはFFEの“かぎりなくバグに近い仕様”で，今度のディスクマガジンで配布するバージョンでは，ちゃんと併用できるように改良されています。

地面は「物体設定」の「追加」で「ji.suf」を，

位置：(0, 0, 0)

拡大：X方向30倍，Y方向10倍，Z方向１倍

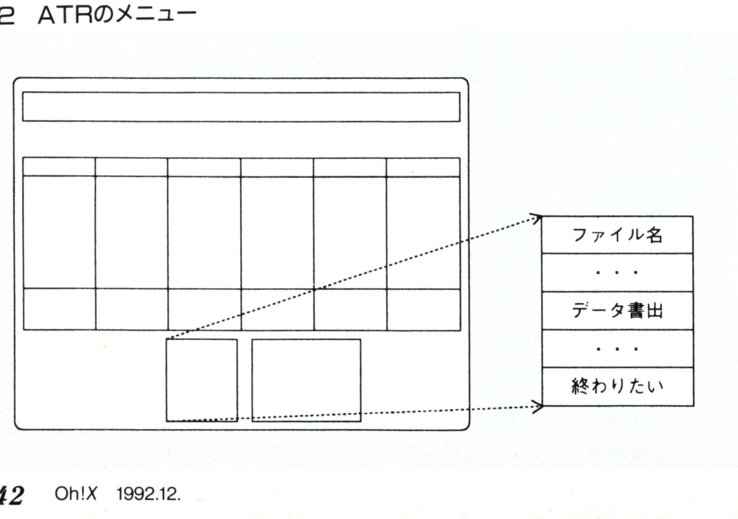
とします（「ji」の大きさが違う方はそれなりに大きくしてください）。

さて，問題の空気遠近法の設定は「背景設定」の「空気遠近法」で，

赤　0.4

緑　0.4

図2　ATRのメニュー



　青　　0.8
　距離　25000

とします。距離というのはマニュアルで解説してある半減距離のことです。距離が異様に長いのは，画角が狭い，つまり視点が非常に遠いためです。通常のように，2000とか5000とかの値にすると，霞がきつすぎて何も見えなくなります。

最後に光源ですが，車の正面やや左から青っぽい光を当てます。「光源設定」で，

　赤　　0.5
　緑　　0.5
　青　　0.8
　ベクトル　(-4,-2,-3)

ぐらいです。

以上で背景のフレームソースができました。「ファイル」で「フレームソース」を選択して，「BACK」として出力してから終了してください。

○車の設定

上記の背景に合成するのですから，視点や光源などの設定は同じです。空気遠近法も同じですが，「TORNADO」をよく見ると，車は背景の柱よりくっきりしている，つまり空気遠近法が弱いような気がします。車を目立たせるための演出でしょうか。

ということで，空気遠近法のパラメータだけは，上記より若干弱め（距離を大きくする）にしておきます。

FFEの起動後，「背景設定」の「空気遠近法」で，

　赤　　0.4
　緑　　0.4
　青　　0.8
　距離　50000

光源は同様に「光源設定」で，

　赤　　0.5
　緑　　0.5
　青　　0.8
　ベクトル　(-4,-2,-3)

視点も「視点設定」で，

　視点　　(29000,-4500,500)
　注目点　(-140,-200,500)
　画角　　4

となります。

「TYRL.SUF」の設定は，「物体設定」の「追加」で「TYRL.SUF」を選択し，位置(-1100,-650,70)に置きます。Z値が0でないのは，以前お話ししたように「TYRL.SUF」制作時のミスです。

「フレームNo.設定」で60フレーム目にしたあと，「物体設定」の「変更」で「TYRL」の位置を(6000,-650,70)にします。

「TYRL」の設定はこれだけで，あとは「X12A.FSC」という名前でセーブしておいてください。

○作画

まず最初に背景を1枚作画します。コマンドラインから，

　REND /A2 HASIRA.SUF JI.SUF BACK.ATR
　BACK.FSC

とします。1枚だけなので，たいして時間はかからないでしょう。曲面はないので，「/g」オプションを付けて，作画を遅くする必要はありません。また，コプロセッサを持っている方はRENDの代わりにRENDXVIを使用してください。

次に車を作画させますが，背景画と合成させながら作画するオプション「/h画像名」を使います。一般に，この「/h」オプションで合成させる背景画は，1枚の静止画の場合と，作画する動画と同じフレーム数の動画の場合がありえます。この判別は，画像名が「画像名＋3桁の数字」のときは動画，そうでないときは静止画となっています。ですから，「BACK001.PIC」を指定すると自動的に動画と判断され，2フレーム目を作画した時点で，「BACK002.PICが見つかりません」というエラーが発生してしまいます。

そこで作画する前に，

　REN BACK001.PIC BACK.PIC

を実行しておかなければいけません。「REN」はCGAシステムのプログラムではなく，Human68kのコマンドでリネーム，つまり「BACK001.PIC」というファイルを「BACK.PIC」という名前に変えています。

準備が整ったら，

　REND /A2 TYRL.SUF TYRL.ATR X12A.FSC /
　HBACK

で作画します。60フレームありますので，時間はかなり

ワンポイントテクニック

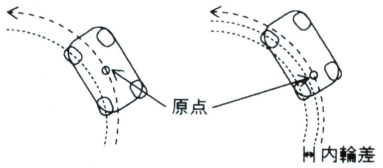
このコーナーではCGA制作において知っておくとちょっとお得なテクニックを紹介します。皆さんもCGAシステムを使ううえでのテクニックや面白い表現方法などがありましたら，ぜひ投稿してください。今回はスタッフ一同をうならせた，愛知県の伊藤さんからの鋭い投稿です。

◎「自動車などの形状デザインの場合には，原点を後輪軸上に置く」

通常の自動車(2WS)の場合，前輪で方向を決め，後輪が動力になります。そのため曲がるときにいわゆる内輪差ができます。しかし，原点を自動車の中心部にしておくと，内輪差のない，妙にタイヤが横滑りしたような曲がり方になって不自然です。そこで，原点を後輪軸上とまでいわなくても，少し後ろに下げておくことで，自動的に内輪差が発生し，自然な曲がり方ができます。

「TYRL.SUF」のように，すでに制作された物体の原点を移動させるには，CADの「File Mode」の「Load」で物体を読み込み，十字カーソルを新しく原点にしたい位置に合わせ，「移動指定」をクリックします。で，十字カーソルを原点に持っていきリターンすると，物体全体が移動します。「固定終了」でこのモードを抜け，再び「File Mode」に入って「Save」します。

図3



1）中心に原点を置いた場合　2）後輪軸上に置いた場合

かかるでしょう。画像データ量も約5Mバイト，ディスク5枚分になります。ハードディスクで作業している方は問題ありませんが，フロッピーディスクしかない方は大変です。

そういう場合はまず5枚のフォーマット済みディスクを用意します。そして，全部のディスクに「BACK.PIC」をあらかじめコピーしておかなければいけません。そうでないと，ディスクを交換した際，合成する画像データがなくなってエラーになってしまいます。やはり本格的にCGAをするにはハードディスクが必要ですね。

○アニメーション

色数は少ないのでCRDを使う必要はありません。

MKTCH X12A001

でタイムチャートを作り，

HANIM X12A

でアニメーションを見ることができます。

ハードディスクがない方は，「MKTCH X12A001」では全部のディスクのタイムチャートはできませんから，自分で書き直してください。

```
.timechart
    x12a [1-12]
.endchart
```

となっている2行目を，

```
    x12a [1-60]
```

とします。「HANIM X12A」を実行した際は，ディスク1枚分の画像を読み込んだところで止まりますので，順番にディスクを入れ替えてください。

○エフェクトをかける

アニメーションをご覧になっていかがですか。前回までの適当に作ったカットとはひと味違うのがおわかりいただけると思います。しかし，文月さんは，さらにひと工夫しています。

このカットは「TORNADO」の作品中に2回に分けて登場します。作品の前半に1～30フレームを，中盤に残りの30～60フレームという感じです。そして，前半での使用においては，この画像を「NEGA」というプログラムで白黒に変換して，回想シーンのような雰囲気を出しています。

NEGAは画像ファイルの色をRGBに分解して(各色0～31)，それぞれを強調，反転させるプログラムで，RGBすべてを反転すると写真のネガフィルムのようになることからこの名前がつきました。白黒フィルム調にするには，RGBの各色の強さの平均値を求めればよいので，

NEGA X12A /OX12B /A=(R+G+B)/3

とします。これはX12Aという動画を入力し，X12Bという出力名でRGBのすべてを，(“A”は“ALL”の意味) Rの強さ，Gの強さ，Bの強さの合計を3等分した値にしろという意味です。

変換された画像は，同様に，

MKTCH X12B001

HANIM X12B

でアニメーションしてみてください。どうです？　なかなかかっこいいでしょう。

しかし，私の個人的な意見として，回想シーンのような雰囲気を出すのなら，完全な白黒ではなく，セピア調にしたほうが渋いと思います。その場合，

NEGA X12A /OX12C /R=(R+G+B)/3+3 /G=(R+G+B)/3−3 /B=(R+G+B)/5

とします。皆さんは，どちらがお好みですか？

“思い”から“感動”まで

本当は，ほかにもいろいろな“表現”がなされているカットを制作しようと思っていたのですが，ページ数も残り少ないので，残念ながらまたの機会にさせていただきます。

もちろん，この“表現”とは，空気遠近法や有色光源を使うことではありません。それは，自分のイメージを伝えるために工夫することです。

1)　「自分はこういったものが好きだ」という“思い”がある

2)　これを人に伝えたいという“イメージ”ができる

3)　伝えるために工夫するのが“表現”

4)　その結果できたものが“作品”

5)　多くの人に見てもらって“感動”を生む

作品制作には必ずこの流れがあります。逆にこれがないようなCGAは作品とは呼べません。単なるデモです。

X68000とCGAシステムがあればCGアニメーションができるのは当たり前で，FFEとAUTOを適当に操作すれば「TYRL」が走り回るカットぐらいは簡単にできてしまうことはすでに実証済みです。でも，それだけでは「TORNADO」に対抗できるわけがないのです。そこにはなんら伝えるものがなく，感動を呼ばないからです。

“いや，適当に作ったカットを友人に見せたら，たいへん感動したぞ”と反論する方もいらっしゃるでしょうが，残念ながらその友人が感動したのは，作品に対してではなく，CGAシステムに対してではないでしょうか(CGAシステム自体，強い“思い”によって開発された立派な“作品”です)。

もちろん，「TYRL」が走り回るCGAといっても，皆さんがスピード感とかかっこよさなんかを追求して，それを表現できれば立派な作品になるでしょう。そしてその過程で気がつくことは，適当にCGAのカットを作るのはとても簡単だが，自分のイメージどおりのカットを作るのはたいへん難しいということです。でも逆にいえば，自分の表現したいものがちゃんと表現できたときはとってもうれしいのです。それが作品制作というものです。

そして，もうひとつわかってほしいことは，いままで

やってきた“打倒TORNADO”という目標はくだらないということです。なぜなら，あの「TORNADO」という作品は，文月さんの“思い”であって，それをマネる必要はないからです。

たとえば，私がTORNADOの形状データを使って，車が走り回るようなCGAを制作したらどのような作品になるでしょう。1990年に，XVIの発表を記念して，私と文月さんで共同制作した「XVIイメージデモ」がよい例です。私自身は車に対してなんの興味も思い入れもありません。ですから，前半の適当にTORNADOが走り回っているシーンなどはまさしくデモでしかありません。しかし，後半トンネルを抜けて夕焼けの街に入ったあたり

読者によるほっとけないほっとこらむ

皆さん，こんにちは。秋が深まり，地方によってはもう冬景色の見られるところもあるのでしょうが，いかがお過ごしですか？

秋といえば，各地で学園祭などが盛んに催されている頃ですね。大阪大学コンピュータクラブでは，毎年コンピュータ占いで“暴利をむさぼって”います。わたしも学生の頃は，ブリッコして受付嬢をやっていました。いまは会社の行事（課内旅行，運動会など）に追われ，「やせるかな？」と期待しつつ，食欲の秋を堪能しているうさ子なのでした。

さて，今月はお便りや質問が多かったので，かまたさんと一緒に回答させていただきます。

〈Aさん〉マニュアルが無事届きました。ありがとう。これからは「打倒TORNADO」を目指して，気長にやっていきたいと思います。そこで質問ですが，X68000を持っていない私はどうすればいいんでしょう？

うさ子：……たしかに気長ですね。

〈Bさん〉夜中に作画して，終了したら自動で電源の落ちる「寝ている間にできちゃったシステム」の導入をお願いしたい。

かまた：電源を落としてくれるプログラム(POFF.Xなど)を使えば，バッチファイルなどで簡単に実現できると思います。でも，私はCRD，MKTCH,HANIMまでを全部バッチファイルにして，ディスプレイの電源だけ落として寝ます。ディスプレイを消せば消費電力は相当少ないはずですし，朝起きてディスプレイをONにすると，すぐアニメーションが見られます。

〈Cさん〉プログラムの名前でKAMA.Xというのがありますが，これはかまたさんが開発したものですか？

うさ子：マニュアルを見ると，「Kantan Anchoku Model Assembler」とありますが，本当の由来は“かまたさんにいわれていやいや開発させられた変な仕様のプログラム”ということだそうで作成者は島さんです。

〈Dさん〉作品制作は自信がありませんが，プログラム開発なら多少腕に覚えがあります。なにか協力できるようなことはありませんか？

かまた：いまほしいツールといえば，フィルムの編集作業のようにアニメーションを編集するタイムチャートエディタ，メタボールのような操作方法を取り入れたCAD，形状データを表示しながらリアルタイムに変更できるATR，ポリゴン専用の高速レイトレーシング，モーションブラーつきのワイヤーフレーム，EPA2の輝かせる機能を独立させて動画に対応したツール，陽炎のエフェクト，古いフィルムのような傷，埃などをつけ加えるツール，地形を生成するツール，草木を生成するツール……。

うさ子：かまたさん，落ち着いてください。まずは画像データにエフェクトをかけるツールやいろんな形状データを生成するツールあたりが簡単ではないでしょうか。とりあえず，なにか自作ソフトを送ってください。また，こちらからご連絡いたします。

〈Eさん〉「　　　　　　　　」

うさ子：「めんどうくさいCGAシステム配布係」宛に，このような白紙のおハガキが何枚か寄せられました。たしかに宛名だけで用件はわかりますが，せっかくだからなにか意見をひと言でいいから書いてほしかったなあ……。

かまた：それ，もしかしてあぶりだしとちゃうか？　ちょっと貸してみ。

……ボッ，メラメラ。アッチャッチャー。

うさ子：あっ，燃やしちゃったら，差出人がわからなくなる！　クスンクスン。

かまた：アッチョンブリケ！

〈Fさん〉しばらく出張で大阪にいることになったから，手伝いにいくことができるぞ！　猫の手が借りたいときは電話してくれい！

かまた：さっそく何度か電話したのですが，連絡がつきませんでした。残念ですが，また次の機会によろしくお願いいたします。

〈Gさん〉友人にCGAシステムを見せてもらい，すぐマニュアルをコピーしようと思ったのですが，さすがに挫折しましたので1部申し込みます。

うさ子：当然ですよね。現実問題として，あのマニュアルをコピーするのは無理というものでしょう。

〈Hさん〉私は会社の両面コピー機を使い，3人の友人と自分のバックアップ用の計4冊，マニュアルをコピーしましたので，その分のカンパを送ります。

うさ子：……ありがとうございます。おそれいりました。

〈Iさん〉AUTO実行時に/Vオプションをつけると，RENDで作画せずに終わってしまいます。

〈Jさん〉AUTOで/Vオプションをつけると作画後，アニメーションしません。

〈Kさん〉FFEの「視点設定」で画面回転(rotx)を行うと，完成予想画面が変になります。RENDで作画してみるとちゃんとした画面のようですが……。

かまた：バグに関するお問い合わせは多いのですが，今のところ，極端に問題のあるバグはないようです。まず，Iさんのご指摘ですが，Jさんもおっしゃっているように，そのような症状は確認できません。もう一度よく確かめて，ご連絡ください。それから，/Vオプションは，形状データをいろんな角度から見た静止画を作成するという意味ですので，アニメーションはしません。アニメーションさせるべきかどうかは仕様的な問題ですので検討します。

Kさんが指摘しているFFEの問題は立派なバグです。FFEはたくさんの方が使っているだけあって，そのほかいろんなバグが報告されています。現在FFEはバグの修正だけでなく，大幅な機能アップが行われていますので，近く（ディスクマガジンと同時に？）発表できると思います。

〈Lさん〉カンパとして，即席チャーハン，牛すき，味噌汁を送る。心して食すように。自衛隊御用達の食料だからのぉ。

かまた：PKOに行っている皆さんも，こういう食事をしてるんだろうか？

うさ子：あの〜，これって，軍事物資の横流しっていいません？

〈Mさん〉予想をはるかにうわ回るものすごいものだったので感動しました。でも，なにぶん私は浪人生。前期はT大ですが，後期は阪大か京大を受験するので，そっちになったらDōGAに参加させてください（可能性小だけど）。

うさ子：「可能性小」というとこが，なかなか自信があるようですね。T大にもTSG(東大理論科学グループ）などのコンピュータクラブがありますから，そちらでも活躍してください。それに別に阪大や京大でないとDōGAのスタッフになれないわけじゃありませんよ。スタッフには，関大や京都芸大の方もいます。

かまた：そういえば，文月さんも横浜から大阪に転勤になったから，完全にスタッフやね。

〈Nさん〉冬休みに見学に行きたいです。連載の隅にでもおよその人数を載せてください。参考にします。

うさ子：“およその人数”とは何の人数のことかな？　プロジェクトルームもあまり広くはないので，あまりたくさんの人数で来てもらっても入れません。差し支えないゲストの数は5名ぐらいまでです。また，プロジェクトルームにいるスタッフの人数は不定（0〜30）です。もし，おみやげの数をお考えなら，あまり気にしていただかなくて結構です。

〈Oさん〉自転車で行けるところに住んでいますが，公務員なもんで表立って参加してよいものかどうか……。

〈Pさん〉マニュアルが送られてきたとき，家族にエロ本かなにかと誤解された。

うさ子：DōGAに対して，なんか妙な誤解をしている方がいらっしゃるようで。

かまた：そういえば，CGAコンテストのビデオを大量に発送したときに郵便局の人に変な目で見られた。いったい，どんなビデオを発送していると思ったんだろう。

〈Qさん〉このたび10年越しの夢がかないまして，晴れてX68000のユーザーになりました(ニコニコ)。CGAシステムの申込書を送ってください。

〈Rさん〉たくさんいないはずの「9月号から買い始めた者」です。X68000もつい最近に中古屋で買ったばかりなので，DōGAでX68000に慣れたいと思います。

うさ子：念願がかなっておめでとうございます。だけど，10年も前からX68000は存在していたかな。お2人ともCGAシステムで楽しみながら，X68000を使いこなしてくださいね。

〈Sさん〉今市を全国的にバカにするのはやめてほしい（今市在住）。

かまた：ごめんなさい。

からは思いが加わっています。その思いの源は，学生時代にひとりでアメリカをヒッチハイクを重ねて横断したとき，ろくに英語もしゃべれないので，ラジオから流れる曲を聞きながらボンネットに映り込む夕焼けをいつまでも見ていた，という思い出にあります。つまり私にとっての車とは，旅のパーツであり，夕焼けのように寂しいものなのです（Graphic Gallery参照）。

このように，まったく同じものを題材にしても，作品はまったく別なものになります。別の人間が作るのだから当たり前です。ですから皆さんにとっても，作品制作の唯一の評価基準はほかの人の作品と比べてどうだという問題ではなく，自分の思いをどれだけ表現できたかということなのです。

まとめ

文月さんもいっているように，X68000とCGAシステムがあれば誰だってCGアニメーションを作ることはできます。誰だって作れはするが，適当に作ったカットを並べてもそれは単なるデモにしかすぎず，“作品”とは大きな隔たりがあります。隔たりがあるとはいっても，文月さんや「TORNADO」は雲の上の存在ではなく，あくまでも私たちと同じ地上の存在であり，そこにたどりつく道もついています。

“デモ”と“作品”の隔たりのポイントは，人に伝えたい“思い”が表現できているかということで，この峠をクリアしないかぎり，「TORNADO」のような“作品”には到達できません。

別にそういった“思い”を最初から持っている必要はありません。道すがら見つけていけばよいのです。また，その“思い”を表現するためのテクニックなんて，これから長い道のりを歩く過程で，体力のように勝手に身についていきます。

ですから，“作品”という世界にたどりつけるかどうかには，たったひとつの問題しかありません。それは，歩き続けるか，途中であきらめるかです。文月さんは，「誰にでもできる」といいましたが，本当に誰にでもできるとは思いません。しかし，歩き続けるかぎり，誰にでも可能であることは間違いないでしょう。

おわりに

9月号で，“7月号を買い損ねてCGAシステムを持っていない”という方のためのマニュアルの再配布のお知らせを掲載いたしました。そんな人はほとんどいないはずだとたかをくくっていたところ，来るわ，来るわ……。最初はちゃんと対応していたスタッフも，大学の試験シーズンに重なったこともあって，オーバーフロー。このままではいけない！

ついに最後の手段として，大阪府下のCGAシステムユーザーに片っ端から電話して，手伝いに来てもらうことにしました（う〜ん，非常識）。9月号で募集した「宛名書き要員」の応募者の方も含め，皆さん快くご協力くださいました。七人の侍（笹間様，松井様，兵頭様，真弓様，矢島様，木野様，黒木様）の皆様，まことにありがとうございました。

次回の発送作業はたぶん来年の4月ごろ（CGAコンテストビデオ）でしょう。大阪府在住の方，いきなり当チームから電話があっても，驚かないでください（赤紙電話と人は呼ぶ……）。

今回で「打倒TORNADOへの第一歩」は終わります。次号では1回お休みをいただいて，1993年2月号から次のシリーズを始めますので，お楽しみに。

当チームおよび，この連載・各コラムに関するお問い合わせ，ご意見などは，下記の住所までお願いいたします。
〒533　大阪市東淀川区淡路5-17-2　102号
プロジェクトチームDōGA「あてにならないアフターサービス係」

CGAマガジン編集部より

さあ，第5回アマチュアCGAコンテストの締め切りもいよいよ迫ってきました。12月31日までにはもう1カ月しかありません。皆さんそろそろ仕上げに入ってください。入選すれば，ビデオになって全国の方々に見ていただけますから，アマチュアCGA作家としては絶好の機会です。ぜひがんばってください。

Q1． 4カット部門に応募しようと思うのですが，その場合，画像データとタイムチャートを送ればよいのですか？　それとも，形状データとフレームソースを送るのでしょうか？

A1． 画像データ＋タイムチャートだけでなく，できればビデオテープに収録して送ってください。ビデオ収録機材がない方はディスクだけでもかまいません。形状データやフレームソースは特に必要ありませんが，将来，ディスクマガジン用のデータとして，提供をお願いするかもしれません。

Q2． 音楽がないと不利でしょうか？

A2． BGMをまったく使わないという演出もありますが，音楽も映像の一部ですから，表現力が狭まるのはいたしかたありません。かといって，著作権上問題のある曲をつけた場合はビデオ化する際に別の曲に代えられるので，トータルイメージがかなり損なわれます。友人などに協力を求め，オリジナル曲をつけるか，Z-MUSICのサンプル曲などを利用してください。クラシックを自分で演奏してもかまいません。また，当方もオリジナル曲を所有していますので，最悪の場合音楽なしでご応募いただき，こちらで適当に曲を入れることも可能です。

Q3． 作品中にお試しシステムのデータを使ってもよいのでしょうか？

A3． まったくかまいません。「TORNADO」の中にも，「人体モデルデータ集」のデータが2つ使われています。しかし，すべてが人のデータで，自作は何もないとかいうことになれば，それなりに減点の対象にはなるでしょう。もっとも，お試しシステムのデータだけでまともな作品ができるとは思いませんが。

来月号は連載がありませんので，これが最後の告知となります。もう一度要点をまとめておくと，

- パソコンによるアマチュアのオリジナルCGアニメーション
- プロもプライベートの機材で，プライベートに制作すれば可
- 機種，ソフト不問。オリジナルソフトも歓迎
- ビデオテープ(Hi-8，S-VHSが望ましい)，8mmフィルム
- 締め切りは，12月31日必着
- 作品に応募用紙をそえて申し込むこと。応募用紙は当事務局に申し込むか，CGAシステムのマニュアルの付録を利用する
- BGMやデザインの点で著作権上問題のある作品は不可
- 1カット部門は15秒以内。4カット部門は30秒以内。ただし，タイトルは除く
- 1カット部門，4カット部門は，処理の都合上，CGAシステムが望ましい
- 問い合わせ先

〒533　大阪市東淀川区淡路5-17-2　102号
プロジェクトチームDōGA内
「CGAコンテスト事務局」

各種ツールを使ったモデリング(1)

文月 涼

前回はトータルイメージの考察を行い，「モデリングしやすい形状とは」というところで話が途切れてしまいました。今月はこれを受けてモデリングに関する講座をお送りします。

■モデリングの概念

モデリングを始める前に，理解しておくべき概念があります。それは基本的なCGにおけるモデリングの手法の違いです。

手法にはいくつかありまして，それぞれ存在する任意の物体をどのように捉えるかによって分類できます。以下はその代表例で，それぞれ任意の物体をどのような感覚で表現するかの概念です。

1) ポリゴン（平面）の集合体として物体を表現する

2) あらかじめ用意された（球体や三角錐といった）プリミティブの組み合わせとして表現する

3) メタボール（濃度分布をもった球体）の配置の構成として表現する

詳細は割愛いたしますが，ポリゴンによるモデリングは厚紙を使って物体を表現する手法，プリミティブは積み木を使って物体を表現する手法，メタボールは近づけるとくっつきあおうとして変形する球（逆もあり）を集めて物体を表現する手法，と考えていただくといいでしょう。

このうちポリゴンは平面の集合体のため，一見するとガタガタのモデルしか作成できないように思えますが，スムーズシェーディングという手法を同時に用いることによって，連続する多くの面を疑似曲面として表現し，モデルの不自然な違和感をそれなりに取り除くこともできます。

現在のCGAシステムでは，1)のみをハンドルしています。したがって，CGAシステムを使ってモデリングを行う場合は，すべての物体を平面の集合体として組み上げていかなければなりません。

つまり，物体をどのように平面に分解するか，という感覚を養うことがモデリング上達への道となります。当然，滑らかな物体，すなわち，球，TORNADOのボディ，人間の体などは単一の平面では捉えにくいため作業も複雑になり，立体に対する特別な感性でもないかぎり，ある程度の図面を用意しなければなりません。

では，複雑な物体を表現しようと思った場合に抜け道はなく，常に泥沼にはまらなければならないのでしょうか。いえいえ，解決策がないわけではありません。CGAシステムに存在するいくつかのモデリングツールを，機能に応じてうまく使いこなすのです。それにはまずそれぞれのツールの特性をよく理解しなければなりません。

ここではその前に，高度なモデリングのうえで欠かせない概念を勉強していただきましょう。

■物体をどう捉えるか

プレモデリングの作業として忘れてはならないのが，「いまから作る物体は単体か，それとも複数の物体で構成された複合体か」ということです。つまり，いわゆる「構造体」なのかどうかなのです。

単体の物体のモデリングの場合は，いきなり「対象をどうやってモデリングするか」という作業にかかれますが，構造体の場合はまず，「対象をどの単位で動かすか」から作業は始まるわけです。なぜなら構造体を構成する各部品は，別々の物体としてモデリングされていなければいけません。

このように，構造体ではまず対象とする物体をモデリングできる単位として分割する作業から始まります。

たとえば飛行機。これは可動な部分がないとすれば，単一の物体として表現できます。では車。最低限でもタイヤは左右に曲がり，かつ回転してほしいものです（お試しディスクのワゴンのタイヤは動きませんが)。こういう場合は，車のボディとタイヤは別個にモデリングしなければならないのです。さらに人体モデルぐらいになると，標準人体フォーマットでさえ，左右対称としても14個のオブジェクトが必要になってきます。

これにともない，物体のモデリングも以前のように「徹頭徹尾CADでやる」のではなく，最近使われるようになってきた「最も簡単な単位でモデリングしておき，KAMAで合成する」手法を用いるといいでしょう。たとえば単一のモデルでありながら複雑な物体である戦闘機では，弾丸型の胴体，主翼，垂直尾翼，尾翼，キャビン，燃料タンクを別々にモデリングしておき，FFEで最終的に組み上げたい形に各物体を配置してKAMAで合体させる方法により，作業を簡素化できるのです。

また，モデリングの簡略化のために，構造体の単位以上に物体を分割してモデリングすることも可能です。

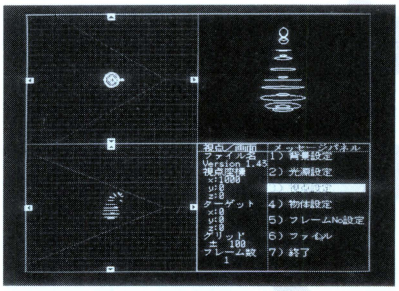
KAMAを使ったこの手法の長所は，「各ツールの機能をいいとこどりする」というものです。CGAシステムの1次モデリングツールとしては，CAD，TAMEN，TUBE，また一度作成された物体を変形する2次モデリングツールとしては，EXPOINT，KAMA，MIRRがあります。これらのツールにはそれぞれ得手不得手があるというか，CADのほかはすべて単機能しかもたないツールばかりなのです。

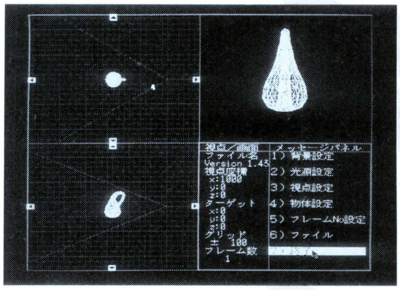
それゆえに各ツールの特化された機能は高く，これを効果的に用いるのがこの手法の主旨なのです。では，特化された機能とはどのようなものなのでしょう。

■簡単らくらくモデリング

以前，「TOYBOX」の記事でもご紹介しましたが，幼少の頃に積み木で遊んだ経験は誰にでもあるでしょう。つまり「TOYBOX」で使われている概念は，より私たちの本能に近い感覚であるといえます。そしてDoGAでも，この感覚に近いものが実現されれば，本能に近いモデリングが可能です。「積み木感覚」ですね。

各ツールの特化された機能とはすなわち，さまざまな形状の積み木を楽に発生させるところにあります。簡単に発生できる形状をツールごとにまとめてみました。一部CADの名称も入っていますが，これはCADがその機能においては特化された機能





TUBEを応用してクラインの壺を作ってみた。まず，断面図となる円を壺の断面となるように配置し，次にTUBEにかける

をもっているという意味なのです。

1) CAD
- 多角柱(角柱作成/回転体)
- 円柱(角柱作成/回転体 ただし,多多角柱として)
- 多角錘(回転体)
- 円錐(回転体/多多角錘として)
- 球(回転体/疑似球面として)

2) TAMEN
- 球(正多角形/疑似球面として)

3) TUBE
- 多角柱(断面図から)
- 円柱(断面図から/多多角柱として)
- 不定形柱(断面図から)

「簡単らくらくモデリング」はこうやって発生させた物体を,FFEで組み上げたい形に配置し,KAMAで合成することで完成します。しかし,この場合でもやはりCADは使用しなければならないでしょう。どのようなモデルでも,最終的にはこのツールによってシコシコと仕上げを行うからです。ゆえに,CADを理解したうえで特化された機能をもつツールを利用するのが,理想的なモデリングの手順なのです。特化された機能をもつツールは単機能なのでつぶしはききませんが,特化されてないCADは多機能であるがゆえにつぶしがきくからなのです。

■複雑なモデリング

では先ほどの形では表現しきれない物体があった場合,どうしたらいいでしょう。

たとえば,いまからTORNADOをモデリングしようとすると,TUBEとCAD+KAMAの複合技に終始すると思います。思いますというのは,私がモデリングを行った時代にはCADしかモデリングツールが存在していなかったからです。

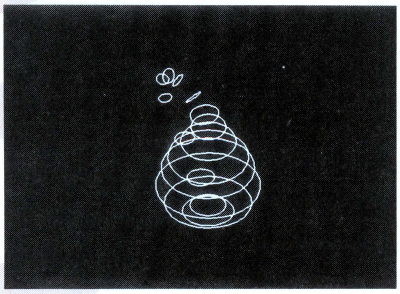
TUBEは先ほども申し上げましたが,多数の断面図から不定形柱(一定の太さ,長さ,方向性をもたない)を作るために存在するツールです。これをうまく利用すれば,どのような流線形の車でも楽に作ることができます。問題点はその断面をいかに作るかなのですが,流線形のすべての面をCADで入力することを考えれば,断面を作成するのは何千分の一の労力でしょう。

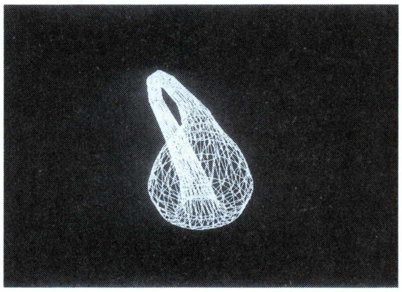
TUBEの特性としてもっておくべき知識は,「なるべく同角数で断面図を揃える」「始点と終点はなるべく揃える」「各面は平行である必要がない」「任意の断面上の点は物理的に単一平面を構成している必要がない」ということです。

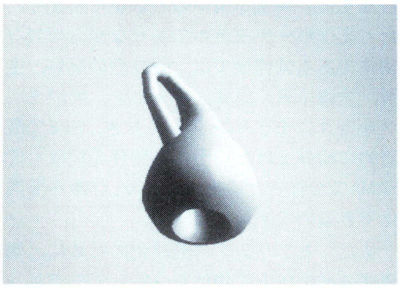
1つ目の角数の問題なのですが,最新版のTUBEでは断面を構成する各面間に角数の制限はないのですが,どちらかというと同角数の面をつないだほうが,美しい筒ができるのです。2つ目で始点と終点を揃えるのは,TUBEは隣り合った断面の対応する点を優先的に結ぶからです。もしこの点がずれていると,筒はねじれてしまいます。3つ目の面の平行性は,断面をうまく配置すれば,とぐろをまいたホースなども表現できるという意味です。

4つ目は理解しにくいと思うのですが,「車の断面を作る場合,必ずしも包丁で切った面を並べなくてもいい」という意味で,具体的にはドアの付け根のラインとフロントガラスの付け根のライン,さらに反対側のドアの付け根のラインをひとつの断面として捉えてもいい,というものです。

それぞれの線がすでに直線ではありませんし,当然同一平面として表現できない点の集合体です。しかしなんらかの手段で断面としてSUFファイルの中に存在していたならば,TUBEは作業対象としてしまうのです。







このように面が交差する場合には,意図した構造をTUBEが理解できずに,誤った立体を作成することも多い。そういうときは一気に処理させずに,数回に分けるようにする。出来上がったいくつかの立体を合成して,目的の立体を完成させればいいのである

CADでも作れない面をどうやって作るのかって? おもむろにエディタを立ち上げて書くんですよ,手で。DōGAのデータファイルは画像データを除いて,すべてテキストファイルですから。

TUBEでは,作成対象とする断面に対して,単一面としての正当性をチェックしていません。SUFファイルの文法が間違ってさえいなければ作業を行ってしまいます。しかし,このときも各面の角数が揃っていないと美しい筒が作成されにくいことは念頭に置いておいてください。

さて,肝心の断面図の作成方法なのですが,いよいよ前回紹介した図面が登場します。作成した図面は各XY,XZ,YZの面間で若干の矛盾はあるでしょうが,要所要所のポイントのチェックはなされていると考えます。つまり各図面間でミラーの付け根の高さが違うとか,ボディの幅が違うなどという矛盾が起きていないか,という意味です。この点をクリアしている図面を使い,まず,どの単位でモデリングを行うかを決めます。

先ほどの物体の分割と違って,今度は面の方向性や対象とする色によって分けるのです。また,その中でも車がX軸の+方向を向いているとすれば,ボンネットからトランクまではX軸に垂直に断面をとると面を作成しやすく,フロントグリルやリアのバンパーなどはY軸に対して垂直(つまりX軸に対して水平)にとらなければ面の表現が困難,というふうに断面の方向性も考えなければなりません。

こういった場合に不可欠な要素となるのが,何度もいっているように「どの面で物体を分割するか」なのです。ただし今度の分割がいままでの例と異なるのは,純粋に扱いやすいデータ量の範囲に単一のオブジェクトの大きさを収めるためなのです。TUBEでのオブジェクト作成は思ったよりも多くのデータ容量を必要とするのです。

前述の例をまとめると,TORNADOでは,

1) フロントまわりを1単位
2) ドア,フロントガラス,ドアの線の継ぎ目までを1単位
3) キャビンを1単位
4) ドアからリアフェンダを1単位
5) リアまわりを1単位

として分割します。この単位でオブジェクトを分割し,TUBEでボディラインを作成していくのです。

では,続きはまた来月。

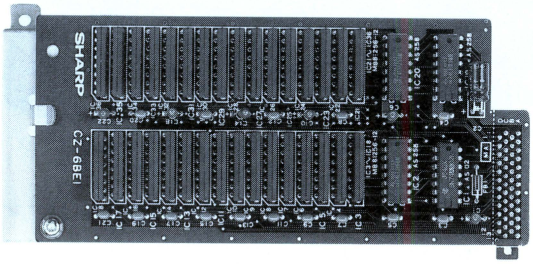
Oh!X 5周年記念

愛読者特別モニタ大募集

今年の6月号では創刊10周年，そしてこの12月号でOh!Xの5周年。16周年とか，256周年のほうがキリがいいという人もいらっしゃるでしょうが，ともかく世間一般的にはキリのいい数字を迎えることができ，たいへんうれしく感じています。これからも雑誌を発行しつづけるかぎり，自然とキリのいい数字を通過していくことと思いますが，続けられるのは読者の皆さんの支えあってのこと。これまでたいへんありがとうございました。これからもよろしくお願いいたします。

今年もシャープさんから盛りだくさんの商品を提供していただきました。記入できる希望番号はひとつだけです。なお，モニタに当選された方々には，1カ月程度の使用期間の後，感想や簡単なレポートを提出していただくことになります。

1 1Mバイト増設RAMボード



(CZ-6BE1B)
28,000円(税別)

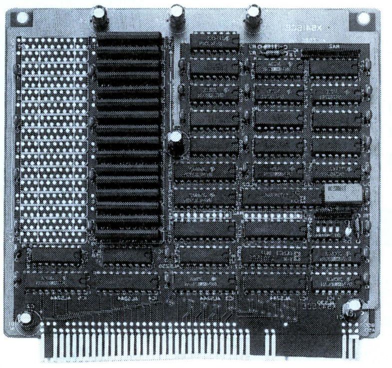
1名

X68000 ACE/PRO（CZ-601C/611C/652C/653C/662C/663C）の本体内に装着する1Mバイト増設RAMボード。

2 2Mバイト増設RAMボード

(CZ-6BE2) 79,800円(税別)

1名



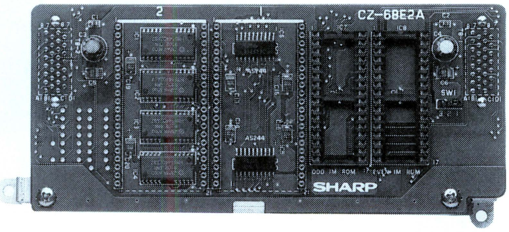
X68000全機種で使える2Mバイト増設RAMボードです。ただし，本体内蔵メモリが2Mバイトになっている必要があります。

3 2Mバイト増設RAMボード

(CZ-6BE2A) 59,800円(税別)

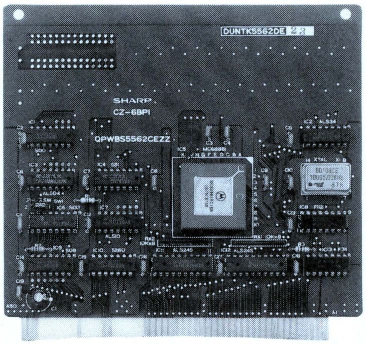
1名

X68000 XVI本体内蔵用の2Mバイト増設RAMボードです。2MバイトRAMモジュール用のソケットも2個装備。



数値演算プロセッサボード

(CZ-6BP1) 79,800円(税別) 1名



5

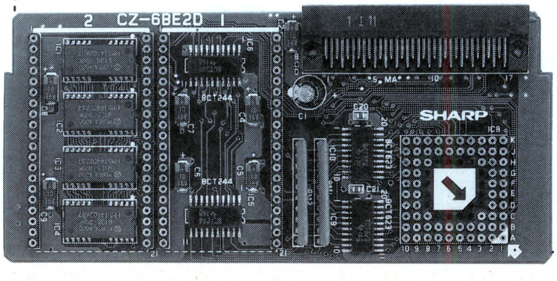
10MHzのMC68881が搭載されたボードです。計算時間を短縮したい方にはうってつけでしょう。

2Mバイト増設RAMボード

(CZ-6BE2D) 54,800円(税別)

4

1名



X68000 Compact本体内蔵用2Mバイト増設RAMボード。2MバイトRAMモジュール用ソケット2個と数値演算プロセッサ用ソケットを装備しています。

6 数値演算プロセッサ

(CZ-6BP2)

45,800円(税別)

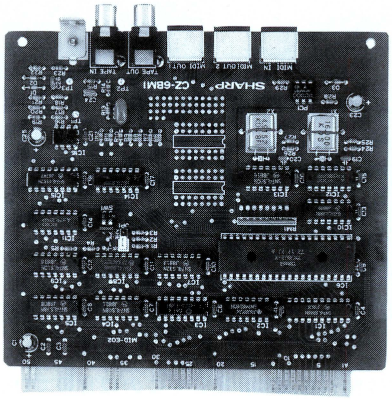
1名

X68000 XVIの本体内, X68000 Compact用2Mバイト増設RAMボードのソケットに装着する16MHzのMC68881。

7 MIDIボード

(CZ-6BM1A)

26,800円(税別) 1名



X68000でMIDI演奏をするのにはこのボードが必要。もちろん, MIDI音源も別に必要です。

8 増設5インチFDD

(CZ-6FD5)

99,800円(税別)

1名

オートイジェクトつき2ドライブ搭載の5インチフロッピーディスクドライブ。ドライブ番号の切り替えも可能。

10 サイバースティック (CZ-8NJ2)

23,800円(税別)

1名

4方向256段階でスティックの位置を読み取るアナログジョイスティック。ボタンもいっぱいついています。

9 システムラック (CZ-6SD1)

44,800円(税別) 1名

X68000やディスプレイを効率よく収納できるシステムラック。コンセントもついているのでタコ足も解消。

11 CARD PRO-68K ver.2.0

3.5/5"2HD版

29,800円(税別)

5名

独自ウィンドウシステムを使用し, キーマクロ, グラフ機能などが追加されたカード型データベース。

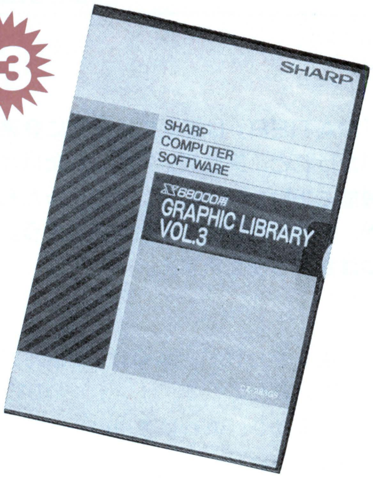
12 EasyPaint SX-68K

3.5/5"2HD版

12,800円(税別)

5名

SX-WINDOW上で動作するペイントツール。スキャナからの取り込みもサポートしています。



13

グラフィックライブラリ vol.3 3.5/5″2HD版

8,000円(税別)

5名

NEW PrintShop PRO-68K, CANVAS PRO-68K, PressConductor PRO-68Kで利用できる，年賀状，クリスマスカード用のグラフィックデータ集。

ダウンタウン熱血物語

3.5″2HD版

8,800円(税別)

5名

14

ご存じ“くにおくん”の登場する面クリ型格闘ゲーム。キャラクターがサンプリング音でしゃべったりもします。

注意　3.5インチ版のみです

15

熱血高校ドッジボール部 サッカー編 5″2HD版

8,800円(税別)　**5名**

こちらも“くにおくん”が登場するスポーツゲーム。タイトルはややこしいけどサッカーゲームです。

16

フロッピーディスクケース

A　5インチ用　**30名**

B　3.5インチ用　**5名**

X68000本体に同梱されているディスクケースを特別にプレゼント。システムディスクが入ってくる例の黒地に金文字のかっこいいケースです。

17

中華大仙

5″2HD版

7,900円(税別)

5名

西遊記の雰囲気いっぱいのシューティングゲームです。仙術を身につけながら敵を倒してエライ仙人に。

10月号プレゼント当選者

1 Easypaint SX-68K　(栃木県)末田正樹　(東京都)島野英男　(岐阜県)小嶋久征　**2** マスターオブモンスターズII　(京都府)大槻典正　(大阪府)岡邑信吾　(岡山県)西博之　**3** ダライアスCD　(兵庫県)矢元章夫　(福岡県)吉田達穂　**4** UFOキャッチャー公式ガイドブック　(大阪府)福森淳　(兵庫県)岡本正和　**5** ヨンセンマン消しゴム　(東京都)本橋純　(京都府)松尾文人　(広島県)神田行男　(敬称略)

以上の方々が当選しました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，雑誌公正競争規約の定めにより，当選された方はこの号の他の懸賞には当選できない場合がありますのでご了承ください。

モニタの応募方法

とじ込みのアンケートはがきの該当項目にすべてご記入のうえ，希望する番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1992年12月18日の到着分までとします。当選者の発表は1993年2月号で行います。

ついに発売，MATIER

Ogikubo Kei 荻窪 圭

「MATIER」は誌面には頻繁に登場していたけれど，実は最近発売されたばかり。機能もだいたいは紹介してしまったんですが，もう一度確認しておきましょう。こいつはかなり使えますよ。

今月はなあ〜んも思いつかんから連載休ませて，っていったら，だめだよん，とかいって，MATIERの紹介するんなら写真撮っておきますからとかいって，こりゃいかん，泣いちゃうぞって笑ったら，2ページでいいから，って話になって，じゃあ2ページ身辺雑事で埋めようかって気もするけど，身辺雑事なんて読まされるほうがたまんないわけで，それはあまりにも両親が悲しむ，じゃなくて，良心が痛む。などと，バリバリとパッケージを開けて，完成品であるところのMATIERをインストールする。

まあ，買った人はとっくの昔に買ったよん，っていうわけだけれども，まだ買ったばかりの人，それから，買ったけど機能がありすぎて何がなんだかわからん，ってな人は，とりあえず，デモを見るように。プログラムディスクのDEMO.BATを実行すればいい。このDEMOがなかなか面白い。何をするかって，広告やパッケージに出ていた，「林檎のできるまで」をリアルタイムで見せてくれるだけのものだが，もう，芸が細かいのだ。MATIERともなると，豊富な機能をどう組み合わせるかが命なわけで，どう点を打てばリアルになるか，デモはそれを見せてくれる貴重なものなのだ。

まず，画面左上にグラデーションペイントをかけた矩形を描く。その上に，自動ブラッシングでブラッシングする。と，自然な感じが出て，CGっぽさが低減される。でもって，お次は球面マッピングする。ちょっとエッジがデコボコするから，背景色でエッジに円を描いて端を直す（このへんが細かい）。まあ，アンチエリアシングするわけだ。

で，メッシュ変形でリンゴの形にする。

こんな感じでリンゴを作っていくわけだね。見るとわかるとおり，必要なのはマウスを動かすテクニックではなく，機能を組み合わせるセンス。

まあとにかく，買った人はデモを見るといい。買ってない人は，買いなさい。こいつは遊べるから。

世間話

最近，パソコンが役に立つかどうかなんてますますどうでもよくなってきて，どれだけ楽しめるか，どれだけ慧眼の手助けをしてくれるか，どれだけ目ん玉の中からウロコを引っ張り出してくれるか，ってそういうことばかり求めている。だいたい，X68000が登場した当時なんて，ハードウェア自体がそうだったから，すごかった。ハードウェア自体にわくわくする要素があったわけ。もう，そういうハードウェアはどこからも出てこないだろうなあと思う。理由はいろいろあるけど，次にそういうマシンが出るとしたら，それはパソコンじゃないだろうなあと。

パソコンの夢，ってなんだろうかと（そんなものはないんだけど，あると仮想してさ）考えるんだけど，それはやっぱり，森羅万象ありとあらゆるものをデジタルデータ化して体内に取り込むことではないかと思うわけだ。あらゆるものをコード化して取り込む。文字と数字だけじゃなくてね。X68000は絵と音をコード化して取り込むという夢をそのハードウェアによって与えてくれたわけだ。いまとなっては絵はフルカラーで，音はステレオ16ビットサンプリング，サンプリングレート44.1kHz，ってとこまできてくれないといやじゃ，なんて贅沢をいっているけど，とりあえず，1986年当時はX68000で何でもできる気がした。

いまはいちおう，ペンの動きをパソコンに取り込もうというのが新しいコード化のひとつの流行りみたいで，つまりペンコンピュータだけど，ペンで描いた軌跡をそのままデジタル化して扱おうとアメリカではのたまっていて，そのデータは「インク」といい，インクテクノロジーって名前で研究されている。

もうひとつのコード化が動画ね。動画。アップルはQuickTimeというパッケージを完成させたけど，本命はMPEG2ではないかといわれている。インテルのDVIってのもあるけど，あれは専用のハードウェアが必要らしい。

で，コード化して体内に取り込む，っていうと，重要なのがどうコード化するか，コード化したものをどう復元するか，ってわけで，これはソフトウェアの領域。CPUパワーにものをいわせて全部ソフトでやっちまおうと。

これが「わくわくするハードウェアは出ないだろうなあ」という理由で，その代わり「わくわくするソフトウェア」はどんどん出てくるだろう。

JPEGはわくわくした。QuickTimeはわくわくした。

個々の製品でいうと，SoloWriterはわくわくした，Morphはわくわくした（どっちもMacintoshのソフトだが）。

そして，MATIERもわくわくした。

やっと話が戻ってきたのだけれど，MATIERは，久々にわくわくしたX68000のソフトなのだ。そこには，パソコンの体内に取り込んだデジタルデータをどう料理してやろう，っていう意気込みがある。1つひとつの機能に意志があり，遊びがある。こういうソフトがもっとあれば，って思わざるをえない。

MATIERのデモを見ると，それがわかる。

MATIERのデモはどうやって実行されているか，っていうと，DEMO.BATの中身を見ればわかる。MATIERに渡すパラメータに“DEMO.MCR”ってのがあるのだ。

この，DEMO.MCRってのは何か。記述されたマクロである。

MATIERでマクロする

マニュアルには載ってないけど，MATIERには超簡単なマクロ機能があるようなのだ。ただし，おまけのようなものだから，期待してはいけないし，途中でとんだとし

ても私の知ったことではない。

このマクロってのは，マウスの動きやキーボードから入力したショートカットコマンドを記録して実行するだけの簡単なもの。あまりに簡単すぎて，マクロ実行中です，とか，マクロを保存します，なんていう表示も何もない。

まず，“(”を押す。すると，記録が始まる。ついで，“)”を押す。記録が中断される。

マクロの実行は，“CLR”キーである。

うまくいかなくても私の知ったことではないが，なんとか動いてくれるようだ。まあ，裏技だな。

裏技とはいえ，マクロの記録開始地点と終了地点を工夫すれば，いろいろと遊べるはずだ。

って，ここで例を挙げられればいいのだろうが，冒頭で宣言したように，何も面白いことを思いつかんのだ。アンアンの真似をして「読者ヌードだ！」なんて思っていたら，「ヌードはだめですよ，ヌードは」とある女性読者にいわれてしまったので，渋々あきらめたわけであるが，それ以来，なんも思いつかんのだ。ああ，貧困。

それはさておき

MATIERを使うにあたって最初に悩むのは，その基本操作ルールだろう。MATIERの面白さはその操作体系にあり，と見る。実に巧妙にさまざまな機能を組み合わせられるようにできているのだ。

まず，右に並ぶメニューがある。ここで，何と何を同時に選ぶか，何と何は同時に選べないか。あるいは，どの機能はどの機能に対して独立なのか。これを掴んでしまえば，あとは見よう見まね，楽しいアドベンチャーゲームの始まりだ。

とりあえず，メニューのいちばん上で色を選ぶ。色は単色とグラデーションがある。グラデーションパレットから色を選んだり，RGBやHSVやCMYのスライドバーから色を選んだりできる。

続いて，透明度。描画するときの透明度を選ぶわけで，下地を透かすこともできるわけで，多くのツールに対して有効である。もちろん，色と同時に選ぶことができる。

続いて，ブラシとペン。これも同時に選ぶことができる。ペンの種類は，いやになるほど多い，っていうか，いくらでも作れる。特にスプラッタブラシ機能にいたっては何でもアリである。ペンに対してブラシ形状が決定されるわけだ。

続いて，セレクション。これは，ペンと排他的関係にある。ペンを選ぶかセレクションを選ぶか，というわけだ。なお，セレクションはつまり閉じた面に対する処理という意味にとらえておくのがいいだろう。

セレクションの右にはマスクがある。マスクに対しても，任意のペンやセレクションが選べる。マスクが選ばれていると，ツールの対象が色ではなく，マスクになる。

その下が，エフェクトである。このエフェクトの多彩さがひとつの特徴であるわけであったりして，エフェクトもセレクションで選んだツールに準じた範囲に働く。つまり，エフェクトが選ばれていればエフェクトが，選ばれていなければ色がセレクションの対象になるわけである。エフェクトには色変換や消しゴム，ぼかしなどなどいろいろと品揃えは豊富である。

その下はエディット。エディットはほかのどのツールにもじゃまされずに単独で働く，ように見えるが，ひとつだけ，エディットに対して有効なツールがある。透明度である。ただし，すべてのエディットツールに対して効くのではないことに注意。

だいたい，以上の組み合わせが主なものだ。ほかにも，ルーペやら（このルーペ機能が秀逸で，拡大して編集すると，「拡大した図に対して1ドットずつ編集を行う」ことができるのだ），仮想画面やらいろいろと並んでいるが，こちらは特殊な領域だ。

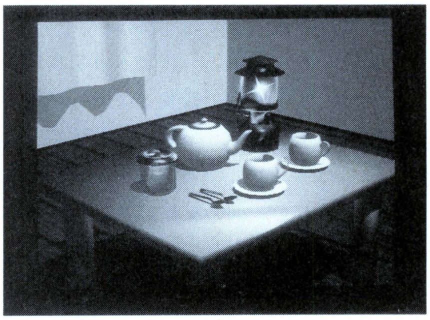
で，ツールを選んだら，メニュー以外のところを左クリックする。すると，メニューが消え，操作するモードに入る。右クリックで，元のメニューのある画面に戻る。

とまあ，以上が作業の基本だ。適当にエフェクト機能を使ったサンプルを作ってもらったので（いや，勝手に作られたのだ），参考にしてほしい。

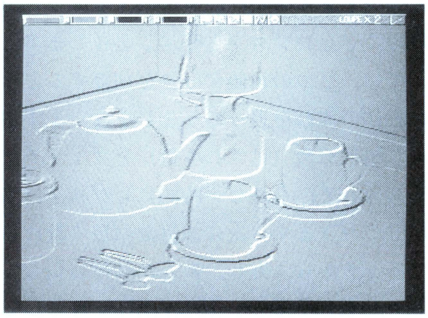
私はといえば，どう組み合わせて，何をどうすれば笑いがとれる絵ができるだろうか，とそればかり考えてもう総白髪中である。せっかく，お手軽小型低価格スキャナ(Macintosh用だけど）まで買ったのに。蜘蛛の巣が張っているのだ。

そういうわけで今月は短かったわけだが，来月はMATIERとプリントゴッコを使ってフルカラー年賀状を作ろうと思わないでもないが，きっとやらないので，ごめんなさい（先にあやまっとく）。

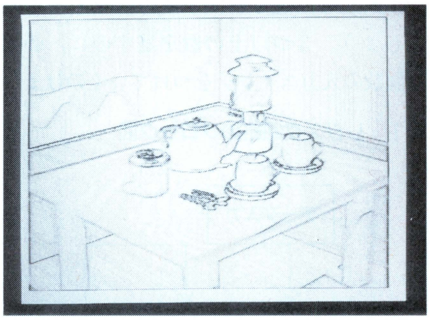
MATIERではモノクロプリンタで印字するときにCMY成分をそれぞれ分けて出力できるから，やろうと思えばできると思うのだが，思っただけではできないのが現実というやつで，なにせ，プリントゴッコがないもので，はははははは。



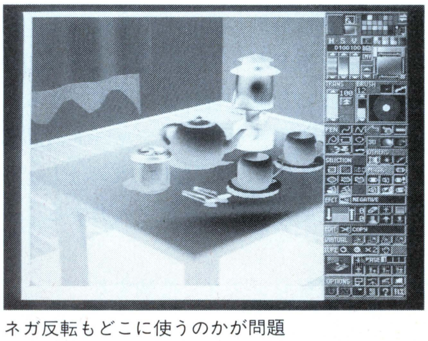
ありがちなフレアの使い方



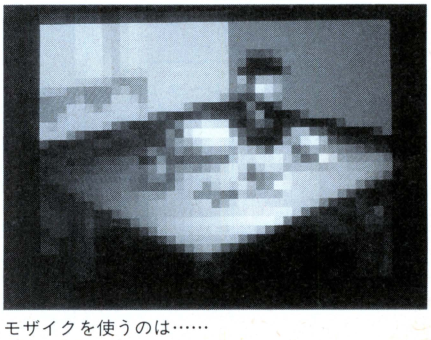
レリーフは使いどころが難しい。壁画などか



輪郭抽出はあとの処理次第で面白くなる



ネガ反転もどこに使うのかが問題



モザイクを使うのは……

コンピュータアーキテクチャ編

レジスタ加算器の設計

三沢 和彦

今月は，ALUで演算された内容を保持するレジスタを考慮した加算器の設計です。いよいよコンピュータらしい話題となってきましたが，そのぶん新しい用語も出てきているので，1つひとつ理解していきましょう。

前回までに2桁（2ビット）の2進数2つの加算を繰り上がりつきで実行する回路を完成させました。2ビット2進数というと0から3までしか表現できませんが，より大きな数の加算を行いたいのであれば，同じ回路を使って桁を増やしていくだけで簡単に設計することができます。ところが，これまで扱ってきた2桁加算器には，コンピュータとして発展させていくにはいくつかの問題点があります。

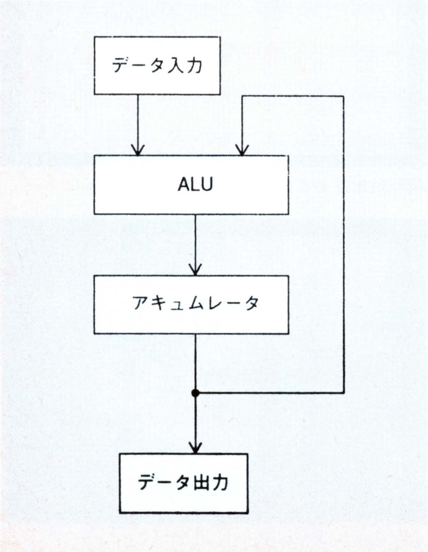
1）演算を施す数値データを入力するのに，2個別々のスイッチが必要となっている

2）計算結果を記憶しておくことができない。すなわち，ひとつの演算が終わって次の演算に移ると，前の結果がどこにも残らない

3）結果的に3つ以上の数を足していくことができない

これらの問題を解決する方針を考えてみましょう。3つ以上の数を足すのに，必要な個数の入力スイッチをつけていくのはまったく愚かな方法です。しかも，TTL ICのパッケージには2入力のALU（算術論理演算部）しかないので，3つ以上の加算をするためには，加算器の最初の回で行ったように，論理回路を自分で組み直さなければなりません。このようにさまざまな手間を考えると，むしろスイッチをひとつにして加算の結果を順次メモリに格納しておき，次の数を足すときにはメモリ内の結果と足していくようにしたほうが，部品数が少なくてすみます。

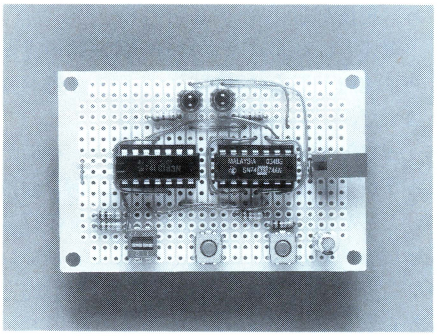
図1　レジスタ回路のブロック図



ここで，用語について注意しておきたいのですが，CPU内部で一時的に値を格納しておく領域のことをメモリとはいわずにレジスタと呼んでいます。メモリとはCPUの外部の記憶領域のことを指す場合がほとんどです。この連載では，しばらくの間CPUの動作をシミュレートする回路を組んでいきますので，値を格納するメモリ領域のことをレジスタと呼ぶことにします。

では，2つの数値データを加算するときの動作の流れを追ってみましょう。まず初めに被加算数（足される数）をスイッチから入力して，それを一時的にレジスタに格納しておきます。次に加算数（足す数）をスイッチから入力し，あらかじめ格納しておいた被加算数と一緒にALUに入力して演算を実行します。その加算結果をまた同じレジスタに格納し直してやれば，その結果が被加算数となります。このように同じ手順で無限に加算を続けることができるのです。以上の手順をフローチャート（計算手順の流れ図のこと）で整理してみましょう。

1）スイッチから被加算数を入力する
↓
2）その値をレジスタに格納する
↓



レジスタ加算器

3）スイッチから加算数を入力する
↓
4）レジスタ内の値を取り出す
↓
5）スイッチ入力とレジスタ値とをALUに入力する
↓
6）ALUからの出力である計算結果を再度レジスタに格納する
↓
3）に戻る

次にいま述べたフローチャートを実現する回路のブロック図を図1に示します。回路はALUとアキュムレータと呼んでいるレジスタとから構成されていて，そこに入出力回路が付属しています。アキュムレータというのは，累積するという意味の英語のAccumlateからきており，計算結果を順次累積していく役割を示しています。とはいえ，機能としては単なる値の格納でしかありません。

このブロック図の示す回路を実際に設計するときの考え方を十分説明していかなければならないと思いますが，その前にまず，レジスタそのものは実際にどういう回路からできているかという点を詳しく説明していきたいと思います。

レジスタとは

レジスタの基本的な機能は，1/0(H/L)のデジタルデータを必要なときに入力し，次にデータを入力するときまで保持し続けるということです。このレジスタは，ハードウェアの面から見ると「フリップフロップ」という回路からできているので，レジスタ回路を理解するのは，「フリップフロップ」を理解することと同じといえます。

「フリップフロップ」とはシーソーの「ギッタンバッコン」の意味で，シーソーのように２つの状態が交互に切り替わっていく動作状態を示しています。

ここでいう２つの状態とは，もちろんH/Lのことです。フリップフロップには機能の違いからRSフリップフロップ，JKフリップフロップ，Tフリップフロップ，Dフリップフロップなどがありますが，そのすべてを説明すると煩雑になってしまうので，ここでは，レジスタによく使われる同期式Dフリップフロップの動作に注目して説明していきたいと思います。参考までに表１にTTL IC規格表から抜粋した同期式DフリップフロップLS74のデータを示します。

同期式Dフリップフロップにはデータ入力端子D，プリセット端子PRESET(PR)，クリア端子CLEAR(CLR)，クロック端子CLOCK(CK)，そしてデータ出力端子Q，$\overline{Q}$とがあります。

ここで，データ入力端子DにHまたはLのデータが入力しているとします。このH/Lは時間とともに変動していてもかまいません。そこで，クロック端子CLOCKにクロックパルスを入力すると，その瞬間にデータ端子に入っていたHまたはLのデータに出力端子Qからの出力データがロックされます。クロックパルスというのは，図２のようにL→H→L（あるいはH→L→H）と入れ替わる信号のことです。LとHとが交互に入れ替わっているので，同じクロックでもL→H→Lとみなすことも，H→L→Hとみなすこともできます。

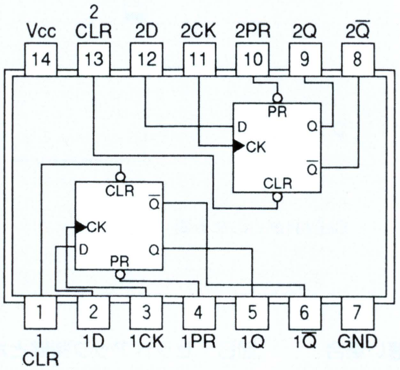
さて，出力データは次にクロックパルスが入ってこないかぎり，入力データがどう変動しようと変わりありません。このように，データの変動とは独立に，クロックという外部制御信号ですべての回路を同時に制御できる仕組みの回路を同期式回路といいます。クロックを入力したときのデータを記憶しているという意味で，これはまさにメモリ（レジスタ）ということができると思います。

このDフリップフロップの動作を時間の流れとともに示した図が図３です（このような図をタイミングチャートと呼んでいます）。このタイミングチャートはデジタル回路の動作の様子を理解するうえで大変便利なものです。

このタイミングチャートを見ながら，ここで重要なことをもう少し詳しく調べてい

表１　LS74規格表

項　目	入　力		出　力	max / min	N	LS	ALS	単位
fmax	Clock		–	min	15	25	34	MHz
tw	Clock	H	–	min	30	25	12	ns
		L		min	37		17	ns
	Preset Clear	L		min	30	25	15	ns
tsu	D		–	min	20↑	25↑	15↑	ns
thold				min	5↑	5↑	0↑	ns
tpd	Clock		Q, $\overline{Q}$	max	40	40	18	ns
	Preset Clear			max	40	40	15	ns
Icc	–			max	30	8	4	mA



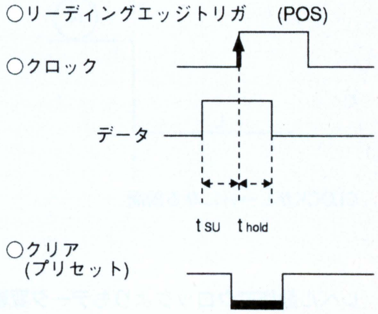


図2-1　クロックパルス

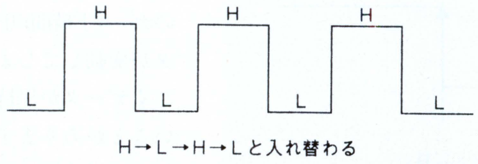
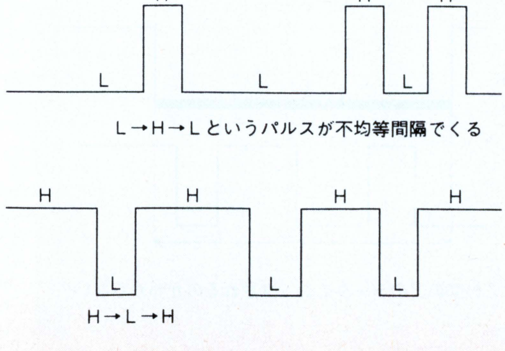


図2-2　不規則なクロックパルス



きたいと思います。注目するものは，

1)　エッジトリガ動作

2)　セットアップ時間とホールド時間

の２点です。

まず，クロックパルスが入ったときに出力データが切り替わるタイミングをもっと厳密にチェックしてみますと，クロックがL→Hに変わる瞬間であることがわかります（図4-1）。このように，制御信号の立ち上がり（あるいは立ち下がり）で動作することをエッジトリガ動作といいます。これに対して，クロックパルスがH（またはL）のときに入力されたデータにロックするレベル動作という方式があります。図4-2ではLレベル動作の例です。

図3　タイミングチャート

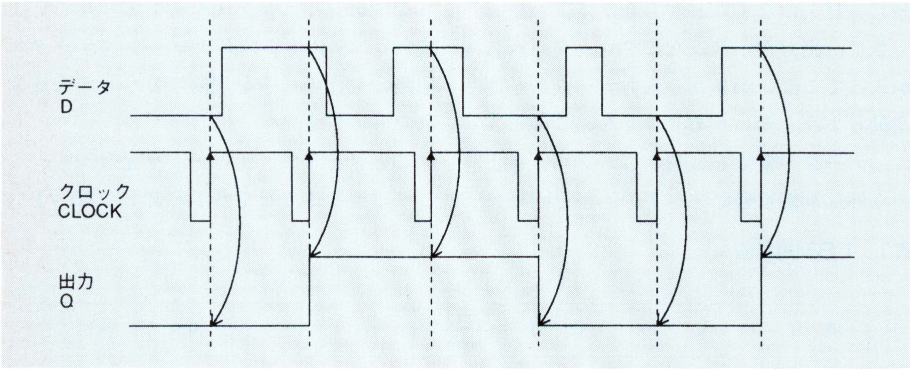


図4-1　エッジトリガ動作

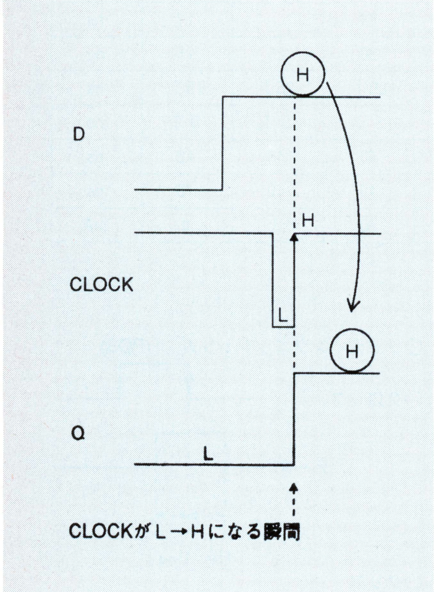
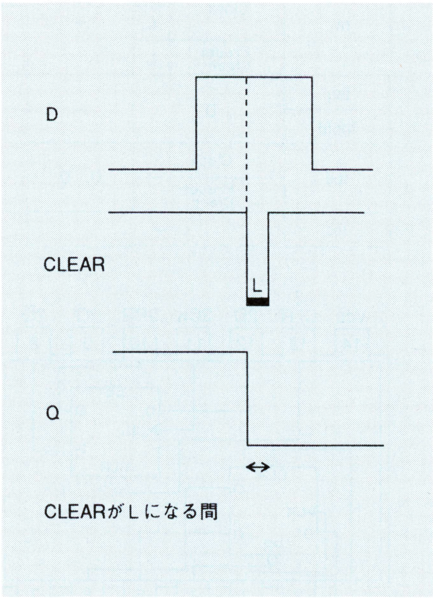


図4-2　Lレベル動作



このエッジトリガ動作の利点は，クロックパルスの幅によらずに正確にタイミングを取ることができるという点です。レベル動作では入力信号のH/Lの変化がクロックのH/Lの変化よりも速く，クロックがLの間に入力が変動してしまうような場合があります。これでは，いったいどこでロックしてよいのかわかりません（図５）。複数のフリップフロップを同じクロックで制御するときには，複数の信号線の時間変化がいろいろあるので，エッジトリガ動作でないと誤動作が多くなるおそれがあります。

クロックのエッジで動作するといっても，安定してデータを取り込むためには入力信号の与え方に条件があります。それが次に説明するセットアップ時間とホールド時間です。これらの意味については，図６を見てもらえれば明らかでしょう。

セットアップ時間とは，フリップフロップに正常なデータを取り込むためにクロックのエッジの瞬間よりも，どれほどの時間だけ前にあらかじめ安定した入力データを与えておかなければならないか，という時間です。TTL ICのDフリップフロップLS74では，セットアップ時間は20ns（1nsは10億万分の１秒）になっています。このセットアップ時間よりもあとにデータが変動すると，クロックのエッジが入ってきたときに入力端子に入っているデータが，正しく保持できなくなることが起こります。

ホールド時間はクロックのエッジが完全に立ち上がった（立ち下がった）あとにさらにどれだけの時間まで入力データの状態を変えていけないかという時間です。LS74ではホールド時間は5nsになっています。このホールド時間以内にデータが変動してしまうと正しくデータが保持されないことがあります。特にホールド時間のほうは今回のアキュムレータつき加算器回路を設計するために考慮しなければなら

図5　レベル動作でクロックよりもデータ変動が速い場合

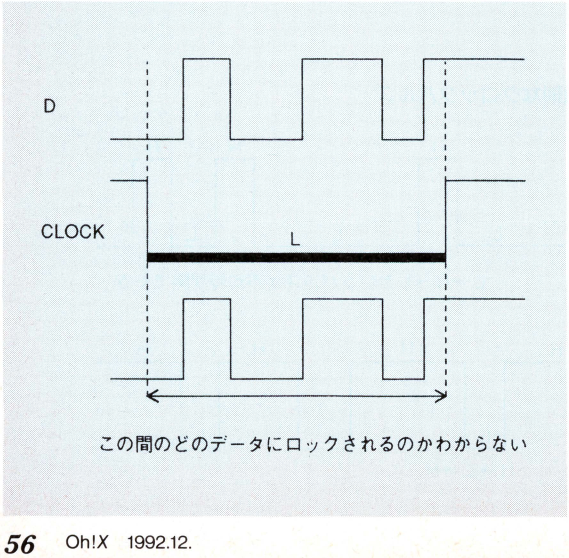
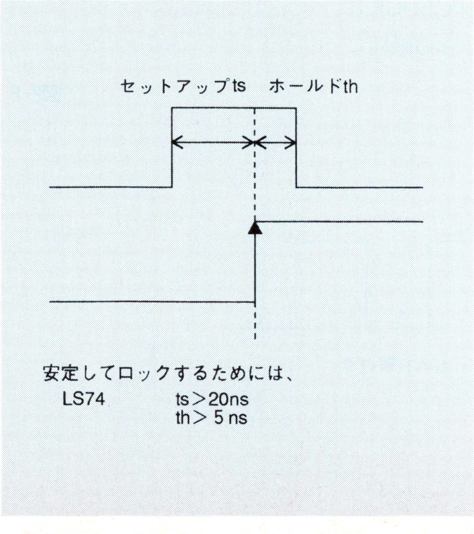


図6　セットアップ時間とホールド時間



なくなってくるので，しばらく心に留めておいてください。

アキュムレータの設計

それでは，同期式Dフリップフロップを使って，加算器回路に重要なアキュムレータを設計してみましょう。加算数と被加算数との2数の入力に対して，加算結果を出力するALUを中心に考えます。ここで，加算数は外部スイッチで入力することにしますが，被加算数はレジスタから持ってくることにします。すると，図7-1のようなブロック図が考えられるでしょう。

次に出力結果をLEDで表示し，その結果を再びレジスタに格納することにします。ここで，出力LEDの位置はレジスタの出力にしなければなりません。これは，ALUの出力はスイッチによる入力値を変えるとそのままダイレクトに変わってしまうのですが，計算結果としてレジスタ内にロックしたあとだと，次にクロックを送らないかぎり，レジスタの出力値は変わらないからです。結局ここまでのところで，図7-2のようになります。

このままでは入力と出力とでレジスタが2つ必要になっていますが，よく考えると出力側のレジスタに格納された前の演算結果を，そのまま次の演算の被加算数にすることに気づくでしょう。ということは，ALUの入力の片方に演算結果である出力レジスタの出力データ（すなわちLEDで表示している結果）を直結してやるだけでよいことになります。最終的には，図7-3のような構成に落ち着きます。そして，この出力データの格納と被加算数データの入力を兼用しているレジスタのことをアキュムレータと呼んでいるのです。

ここで計算結果のデータが出力されている配線ラインに着目してみます（図7-3の太線部分）。この回路を図7-4のような書き方に書き直してみると，このラインにはこの加算器における主要なデータが流れており，ここからALUの入力やLEDの出力を引き出しているのだといい表してもよいことになります。このようなデータの幹線を「バスライン」と呼んでいます。

バスラインというのもコンピュータの構成を考えるうえで非常に大切な概念です。このブロック図では入力スイッチからのデータがバスラインと別のラインを通って入ってきていますが，もし本当にきちんとバスラインを設定するのであれば，スイッチ入力も同じラインから受け取るようにしなければなりません。しかしながら今回の回路では，まだこの重要性がはっきりするほどではないので，今後回路がもう少し複雑に

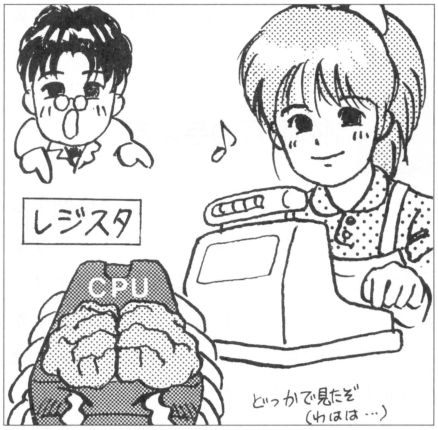


illustration:Y.Kawahara

図7-1　加算器回路のブロック図

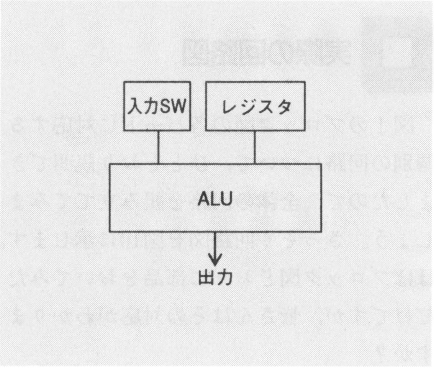


図7-2　レジスタを考慮したブロック図

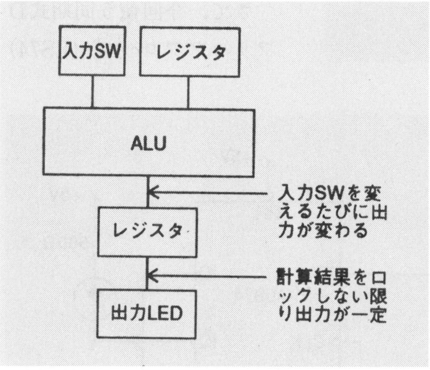


図7-3　最終結果

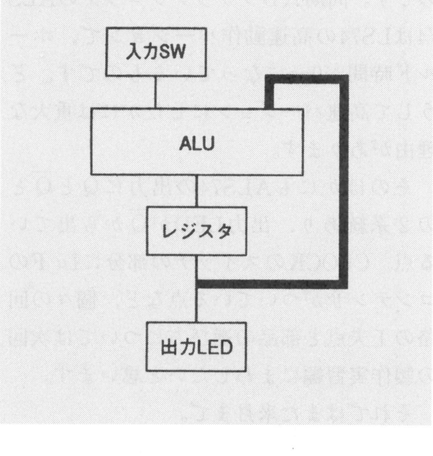


図7-4　バスライン

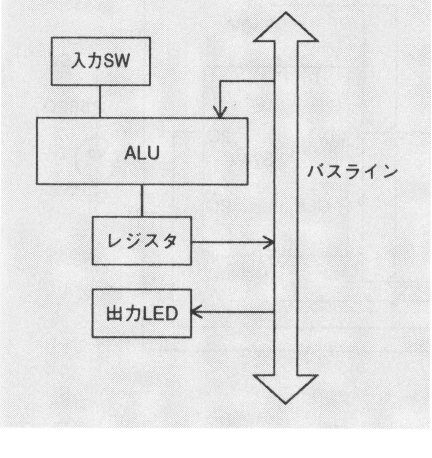


図8-1　スイッチOFFでH信号が発生

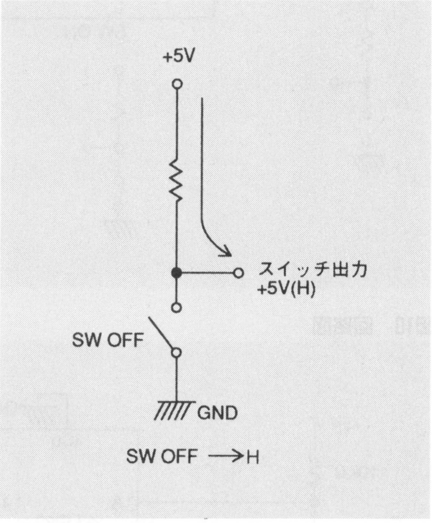
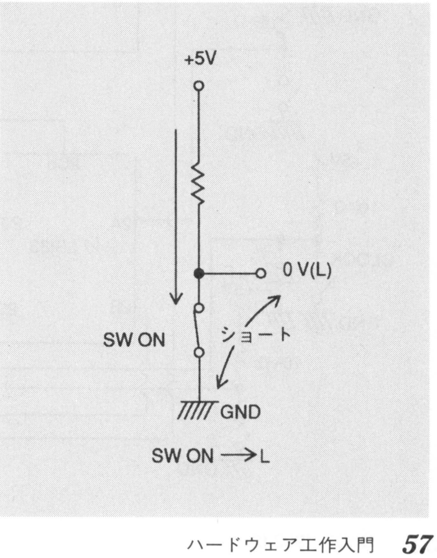


図8-2　スイッチONでL信号が発生



なってきたときに改めて説明すると，重要さが納得できると思います。

制御信号の与え方

先ほどからレジスタにデータを格納するには，クロック信号によって行うというようなことを書いてきましたが，このクロックはどのように与えればよいのでしょうか。

今回使う同期式Dフリップフロップには CLOCK端子があります。この端子にL→H→L（あるいはH→L→H）という信号を入れればよいのです。ここで，クロックという言葉から連想されるのは，図2-1のように時間的に規則正しい信号のことだと思うに違いありません。しかし，実際はL→Hの立ち上がりが検出されさえすれば，時間的に不規則でも同期式Dフリップフロップの制御信号として機能するのです。今回は，押しボタンスイッチによって人間がクロック信号を発生させることにします。

押しボタンスイッチでクロック信号を発生させる回路は図8のようなものです。押しボタンスイッチには，跳ね返り式という押して手を離すとばねによって元に戻る形式のものを使います。

まず，通常の状態では，スイッチはOFFになっているので，図8-1のように出力がH状態になっています。次にスイッチを押してONにするとGNDとショートして出力はLになります。手を離すとまた元に戻って，Hになります。結局通常はHの状態になっていて，スイッチを押すことによって，H→L→Hの信号を作ることができるのです（図9）。

さて，今回使う同期式Dフリップフロップ（LS74）ではL→Hの立ち上がりのときにデータが取り込まれるので，ボタンを押して手を離した瞬間にデータがロックされることになります。これはあとで回路の動作チェックをするときにポイントとなるので，覚えておいてください。

図9　スイッチでH→L→Hクロックを発生

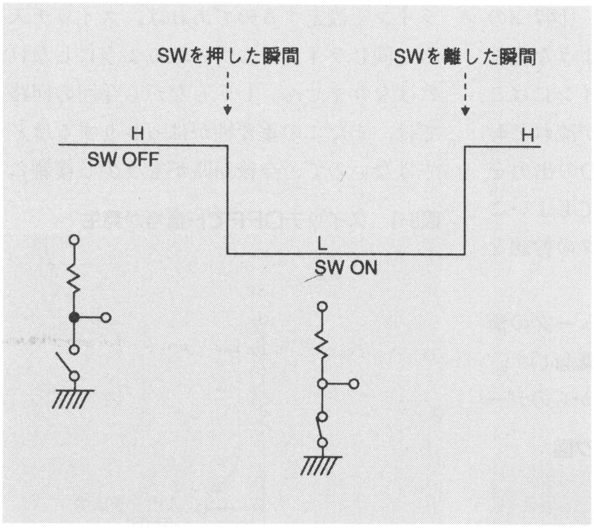
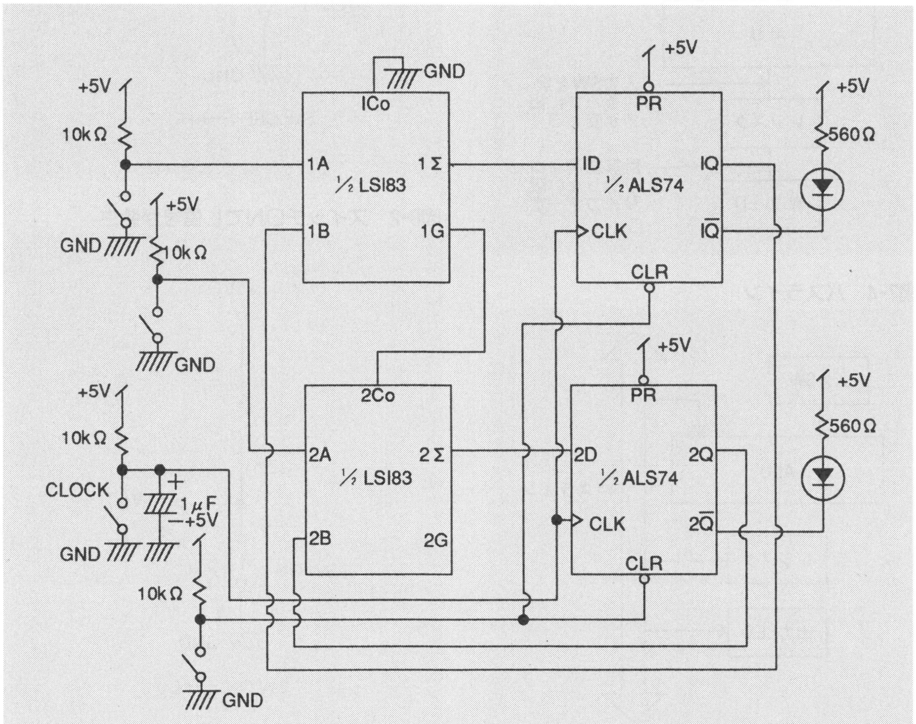


図10　回路図



このクロックボタンは加算器でいえば，プラス（＋）キーの役割をしていることになりますが，実はこのクロックのほかにもうひとつ制御信号が必要です。すなわち，最初に計算を始めるときにレジスタ内をクリアするキーに対応する信号です。この加算器におけるALUでは，入力スイッチの値とレジスタの中身とを加算してしまうので，最初にレジスタ内を0にリセットしておかないと正しく計算が始められないことになります。

これにはDフリップフロップのCLEAR端子に信号を送れば実現できます。今回は，これも押しボタンスイッチによって人間が指令を出すことにします。回路は上で述べたクロックと同じものです。ただし，今回のLS74では，クリア信号はLレベルで動作するので，今度はスイッチを押している間（実際には押した瞬間）に実際にレジスタ内がクリアされることになります。

実際の回路図

図1のブロック図の各パートに対応する個別の回路について，ひととおり説明できましたので，全体の回路を組み立ててみましょう。さっそく回路図を図10に示します。ほぼブロック図どおりに部品をおいてみただけですが，皆さんはその対応がわかりますか？

ALUのLS183は前回の加算器と同じものです。同期式DフリップフロップのALS74はLS74の高速動作バージョンで，ホールド時間が0nsになっているものです。どうして高速バージョンにしたかには重大な理由があります。

そのほかにもALS74の出力にQと$\overline{Q}$との2系統あり，出力LEDが$\overline{Q}$から出ている点，CLOCKのスイッチの部分に1μFのコンデンサがついている点など，個々の回路の工夫点と部品の選び方については次回の製作実習編にまわしたいと思います。

それではまた来月まで。

版下作成支援ツールY-300A

X68000用の版下作成ツールの登場です。あまり馴染みのない呼び名かもしれませんが，DTPソフトのようなものと思ってください。なお，このページはY300-Aを使用して出力してみました。

版下作成ソフトとは

版下作成といっても一般の人には馴染みがないかもしれない。世間でいうDTPソフトと思えばいいだろう。ただし，PageMakerのようなものを想像するとちょっと違和感があるだろう。むしろ**ドロー系**のグラフィックツールに文字処理機能が加わった感じのものだ。基本的に１ページ単位で版下を作成していくツールである。

図形処理機能はかなり豊富というか，特殊な機能も備えている。任意の点列で与えられた線に対する平行線を作成できたり，線種に鉄道線というものがあったり，さらにＪＲタイプと私鉄タイプの指定ができたり。ドロー系のツールとして見ても面白いものがある。

文字処理関係では字詰め，段組数，字間，字送り，段送り，**書体**，長体，平体，*斜体*，回転体，**文字の太**さなどを指定できる。もちろん，文書の任意の位置の設定を変更できる。半面，字詰め設定で半角の大きさが固定になっていたり，カーニング（字間を詰めること）がサポートされていなかったりという不満点もある。うまくやればこのような印刷物も手軽……ではないが，作成できる。Y300-Aの分厚いマニュアルはY300-Aで作成されているようだ。

図版の表示/出力に内蔵のビットマップフォントは一切使わない。すべてベクトルフォントで行っている。これにはZ'sSTAFF PRO-68Kと同じものが使用できる。また，標準で細線タイプのものが用意されているので困ることはない。見落としがちなのは，ちゃんと半角文字にも対応していること。SX-WINDOWでは半角文字はビットマップフォントだったことを思い出してほしい。さらに全角と半角で違う書体を指定することもできる。

単位系として**ミリ，ポイント，級数**の３種をサポートしているのは注目に値する。出版業界以外ではあまり気にする人もいないだろうし，実際気にしなくてもよいものだが，PressConductorのように一見して出版についてなにも知らない人が作ったとわかるソフトとはちょっと違う。

現在，DTPソフトといえば，MacintoshIIだが，海外産のDTPソフトはすべてポイント指定なのだ。ポイントというのは海外の活字の単位で１ポイント＝0.3514mmに相当する。日本での活字の単位は級数で，１級＝0.25mmに相当する。

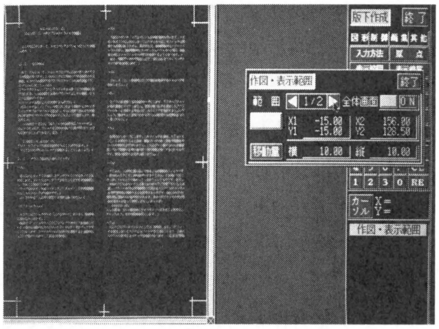
ジャストシステムの「大地」を除いて級数が使えるDTPシステムというのは少ないのだ。変換は簡単なのに対応していない。これは日本語の出版に関する既存のノウハウが反映されていないということだ。DTPの最高峰であるMacintoshIIのソフトでさえ評判が悪いのはこのへんに原因がある。

処理速度は？

さて，Y300-Aだが，処理速度はあまり速くはない。DTPソフトの処理速度などは最初から期待するべきものではない。しかし，処理の重さによるものというより，まだまだシステムの構成を変えるだけで速くできるはずだという印象を受ける。

解像度が足らないのでアナログ多階調を使って縮小文字を表現しているのだが，そのあたりの処理の弊害も見受けられる。画面上のフォントが使い回しできないので，１文字ごとにディスクをアクセスするのだ。単純なフォントキャッシュは付加できないにしても，途中の段階でフォントキャッシュを組み込むことはできるだろうし，再表示ももっと利口にできる。表示時のみ量子化して１ドット以下の誤差を無視するなり，アナログ多階調を行わない高速モードもほしかったところだ。なにかやると全体を描き直すあたりはSX-WINDOWに乗っていればもっと高速だったろうなと思わせる。

いずれにしても，待たされる処理が多く，



１ページが縮小表示される

途中でキャンセルできないものがほとんどなのは問題かもしれない。

使い物になるかな？

この原稿の出力で想定した書体指定は，

タイトル部	44級ゴナB
リード文	12級ゴナDB平１
本文	12級MMOKL
小見出し	16級ゴナDB右長斜３

というOh!Xではお馴染みのものだ。さすがに写植文字と同じものは揃っていないので，書体倶楽部の*フォントを***調整**して使用してみた。そのほか，こわこわ[illegible]のものを加えている（読みにくいだけか？）。

流し込む文章の作成にはかなりの手作業が入っている。一度ワープロで作った文章を最終的にエディタ上で調整していく。なお，プレビュアとしてSX-WINDOWのエディタ.Xを使用した。レイアウトも一度紙に描いたものを参考に指定している。試行錯誤するより確実に楽に作業ができる。なんだかDTPとはいいがたい部分もあるが，出力用と割り切ってみるとPressConductor PRO-68Kよりは細かいことができることがわかる。

発売版は今回試用したバージョンから文書処理あたりが全面的に改訂されることになるらしい。詳しいことは，追ってレポートしたい。　　　　（縮小率７１％）

X68000用　５"２HD２枚組　　価格未定
マグマソフト

SX-WINDOW対応

追いかけっこゲーム

Ishigami Tatsuya 石上 達也

どこかで見かけたようなSX-WINDOW対応の「追いかけっこゲーム」です。ぜひ皆さんのデスクトップで遊んでみてください。自分で迷路やキャラクタを変更するための手順についても解説します。

過去はいつも美しく、未来はなぜか懐かしい。

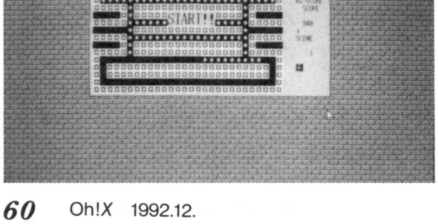
昔の人はうまいことをいったもんだ。

さて，MS-Windowsにはエンターテイメントパックなるピコピコゲーム集が発売されています。なかには，うおぉぉっと，うなってしまう，とてもピコピコとは呼べないような力作から，思わずほほえんでしまうようなピコピコまで，バラエティに富んだゲームが収められています。土俵がWindowsに変わっているのに，ひと昔前の8ビットパソコンのBASICを使った懐かしい匂いがプンプンします（なにゆえ，このようなピコピコゲームをウィンドウ環境で行うとガゼン楽しくなるのかは，The WINDOWS 8月号をご覧あれ)。

そんなわけで，SX-WINDOW用に追いかけっこゲームを作ってみました。どこかで見たような気がするのは，やっぱり懐かしさでしょう。

ルール（遊び方ともいう）

まず，あなたは「2」「4」「6」「8」のキーで人間マークを操ってください。それぞれのキーが下，左，右，上に対応しています。ヘビからうまく逃げながら迷路内に散らばったエサをすべて食い尽くします。



○を食べると，一定時間だけ人間とヘビの強さが逆転します。いつもはヘビにおびえている人間ですが，このときだけはヘビよりも強くなって，逆にヘビを退治することができるようになります。このとき，ヘビは人間に恐れをなして一時的に体が青くなっています。

と，ピコピコゲームの原則に沿ってゲームは展開されていきます。うん，レトロな感じ。

フルーツは出てこないのですか？　という素朴な疑問は却下します。自分でプログラムを改造してみてください。

プログラム

リスト1がゲームの本体です。いつものとおり，といっても私がSXのプログラムを誌上で発表するのは今回が初めてですが，スケルトンは中森氏の連載「よいこのSX-WINDOW講座」より拝借させていただきました。そこらへんの説明は，中森氏の連載を見てください。

ただし，開発キットなるものがシャープからリリースされ，いままで不明だったC言語からのメニューマンの呼び出し方がわかったので，中森氏のオリジナルな方法から，シャープの流儀に倣うように変更しておきました。

残りの（というか主役なのだが）ゲーム部は以前FuzzyBASICコンパイラのサンプルでつけた「PACくん」から，ほとんどのルーチンを持ってきました。

“SWORD”上の「PACくん」は，いうまでもなくシングルタスクだったので，好き勝手なことをできましたが，SX-WINDOW上ではいろいろな制約を受けるようになります。

いちばん顕著な制約が「あまり長い間，MPUを占有してはいけない」という，イベントドリブン式のマルチタスクシステムでは，当たり前といえば，当たり前すぎる制約でした。

たとえば，ゲームスタート時に画面の中央に「START」の文字を一定時間出しますが，コンピュータにこの「一定時間」を作り出してもらわなければなりません。シングルタスクのシステムでは，ただ空ループを回しておき，時間の過ぎ去ってくれるのを待つことで解決できました。空ループが終わったら，文字列を消してメインルーチンへ復帰すればよかったのです。この方法でなんの不都合も起きませんでした。

しかし，SX-WINDOWではこのようなことは許されません。「PACくん」が空ループを回して時間をつぶすようなことをすれば，その期間，ほかのタスクを止めてしまうことになります。

しかたないので，SystemCountという変数を用意し，これを用いることにしました。アイドルイベントが発生するたびに，この変数は0に近いほうに1ずつ変化していき（つまり，値が負のときは1増え，正のときは1減る)，0になったときに，ゲームのメインループをひと回りさせます。メインループをひと回りすると，SystemCountにはSystemWaitという変数の値がコピーされるようにしておきます。

メインルーチンが，次に実行されるのは，アイドルイベントがSystemWait回だけ起こったあと，つまり，SystemCountが再び0になったときです。このSystemWaitの値を変えられれば，ゲーム全体のスピードが変えられるようになるのですが，そこ

まで凝ったことはしていません。暁子.Xなどを立ち上げて，MPUパワーを分散するなどして，スピードを調節してください。

SystemCountが負の数から，0になったときには，迷路を描き直します。0になったあとは，前述のようにSystemWaitの値が代入され通常の処理に戻ります。このことによって，待機中に書かれていた文字列の消去を行います。文字列の描写を行ったあと，その文字列をどのくらいの期間，表示しておけばよいのかを負の値で表し，SystemCountに代入してやります。

蛇足ながら，SX-WINDOWではプログラムはリエントラントに書きなさい，という点にも苦労したのですが，結局挫折してしまいました。挫折してしまいましたので，ぱっきり，OBJC型（リエントラントでないプログラム。シェルから起動されるたびに，同じプログラムがメモリ上にしつこく展開される）になってしまいました。

いやぁ，正直は，実にすがすがしい，はぁはぁ。

噂によると，グローバル変数をすべてA5レジスタ相対で参照するというスグレモノのGNU-Cコンパイラが存在するそうなので，リエントラントでないことに我慢できない方はそちらのコンパイラを使ってください。

自分でいうのもなんですが，68行で定義しているMAP型変数を，メモリマンも使わずいきなりグローバル変数として宣言してしまうところなんか，けっこう悪いことだと思われます。

キー入力について

このプログラムでは，キー入力をキーダウンイベントで取り込み，それを保存しておき，アイドルイベント時（つまりはゲームのメインルーチンを回すとき）に参照するという方法をとるつもりでした。

文末が「つもりでした」となっているのは，実際にはなっていないからで，ちょっと不都合が起きてしまったのです。

キー入力をキーダウンイベントで拾うことはできました。これは，当然です。ショートカットキーの処理などで使いますから。で，当然のことながらこのままでは，オートキーリピートがきいてしまいます。つまり，画面上の人間をつー，っと動かそうと思っても，実際には，つっつっつっつー，と動いてしまうのです。わかりますか？これでは，ゲームになりませんので，1回ごとにキー入力をクリアしないで，キーダウンイベントが起きたときに，その文字を拾い，キーアップイベントが起きたときに，その情報を破棄するという方法をとってみました。リスト1中の，

```
procKEYUP() {
 inkey = 0;
}
```

というのが，キー入力情報の破棄部分です。

でも，これでは，キャラクタは止まらないのです。どうやら，このルーチンがうまく働いていないようです。

大昔のFM-7のゲームのように，自分のキャラクタの移動を止めようと思ったら，移動に関係のないキー（たとえば「5」など）を押してやらなくてはいけないのです。どうもキーアップイベントというのは，このような目的に使えるイベントではないようです。

今回は，なんとなく悪いことなのではないかなぁ，と思いつつもIOCSコールのBITSNSを使って直接キーマトリクスを読み込みにいっています。

プログラムの先頭でKEYUP_EVENTというマクロを定義することにより，このキーボードマトリクスを読みにいく部分を削ることができるようになっています。キーアップイベントの正しい使い方がわかる人は，直しておいてください。

コンパイルの方法

このゲームはソースリストの中にキャラクタデータを埋め込まずに，勝手にその日の気分でキャラクタを作り出すことができるようになっています。本当は，リソースマンを使って，ポイッと追いかけっこゲームのウィンドウにリソースファイルを落とせば，ヘビが怪物になったり，人間がコミカルなキャラクタになったり，できるようにすれば，最高です。最高なのはわかりきっているのですが，あまりリソースマン方面に手を出したくなかったのと，不精な性格がたたって再コンパイルによりキャラクタおよび迷路を変更するようになっています。

リスト5が標準完成品なのですが，再コンパイルしたい人はSX-WINDOWに付属のパターンエディタを用いて，

1）　人間
2）　ヘビ
3）　ヘビの弱ったところ（今回は，2のパターンを青くしただけ）
4）　迷路の壁
5）　人間がヘビに捕まったパターン
6）　人間が元気になるエサ
7）　通常のエサ
8）　なにもないキャラクタ（消去パターン）

を，12×12ドットの大きさで作成します。たとえ白黒しか用いないパターンでもPAT4タイプでセーブします。ちなみに，このときのファイルサイズは，1キャラクタにつき104バイトです。セーブする際のファイルネームは上からhito.pat，hebi.pat，ijike.pat，kabe.pat，kuware.pat，power.pat，esa.pat，erase.patをそれぞれ使ってください。

キャラクタパターンの作成が完了したら，迷路のデータを作成します。通常のテキストエディタを用いて，20×20くらいの大きさのものを用意します。迷路の壁の部分は「*」，通常のエサを置くところが「.」，人間が元気になるエサが「O」（アルファベットのオーです），ゲームが始まったときに人間が出現する場所が「C」，同様にヘビが出現するところを「M」でそれぞれ表してください。後ろの2つは省略してもかまいません。そのときは，迷路の中央やや下側に人間が，やや上側にヘビが出現するようになります。

このとき，タブコードなどを使ってはいけません。そこまで手は込んでいませんから。参考までに，私の作った迷路はリスト2のようになっています。

キャラクタパターンと迷路のデータが入力できたら，リスト3に示すプログラムを使って，アセンブラにかけられるようなかたちにします。カレントディレクトリに，キャラクタパターンと迷路データのファイルをすべて放り込んで，コマンドラインから，

```
A>PATMAKE
```

のように実行します。すると，しばらくして，カレントディレクトリ上に，PATTERN.Sというファイルができあがっていること

でしょう。これを,

A>AS PATTERN

などとして，アセンブラにかけて，PATTERN.Oにしておきます。

この手順が嫌な人はリスト4に私の作ったファイルを載せておきますので，これを黙々と打ち込んで直接いじっていただければ，パターンエディタを立ち上げたり，リスト3を打ち込んだりする手間暇が省けます。本当は，この手間暇をかけることが楽しいのだと思いますが。

それはともかく，どんな手段を用いても，PATTERN.Oというファイルが手に入った人は続けてリスト1を打ち込みます。これがゲームの本体となります。

これが終わったら,

gcc -O -fomit-frame-pointer
-fstrength-reduce
pack.c PATTERN.o
sxlib.l doslib.l iocslib.l

のようにコンパイルをして完成です（リスト5)。ご苦労さま。

リスト2 (114バイト)

```
0000   20 DC 2D 6C 68 35 2D 4F   : AE
0008   00 00 00 F1 01 00 00 FC   : EE
0010   A2 DC 18 20 01 07 6D 61   : 8C
0018   70 2E 64 61 74 21 70 48   : B0
0020   00 00 00 28 4A 97 66 C8   : 37
0028   39 FF 5F EF 73 38 37 80   : E8
0030   0E 64 F2 70 62 D4 91 6E   : 09
0038   0D A2 2F FE 98 00 34 DA   : 82
0040   6E A7 BE 34 16 4A EC 10   : 63
0048   CC 26 6D 2E 87 F9 91 0D   : AB
0050   82 E3 CA B2 30 EE D4 BD   : 90
0058   23 CB 22 45 D9 92 F7 2C   : E3
0060   06 5B 7A E4 E9 AD 91 76   : 5C
0068   3E 2E 23 DD 56 BE 4F 15   : E4
0070   D2 00 00 00 00 00 00 00   : D2
0078   00 00 00 00 00 00 00 00   : 00
--------------------------------------
SUM:  7B EF DD 7D 7A 2E 94 15   3042
```

リスト1

```
/*
    追いかけっこゲーム
    Programmed By T.Ishigami

    スケルトン Ver.2.0
【改版履歴】
     Jul.11,1991 新規作成
         (C) 中森 章, Jul.11, 1991
*/

#include <stdio.h>
#include <FCNTL.H>

#define __POINT_T   /* point_t 型を使う */
#include "sxlib.h"
#define FALSE   0
#define TRUE    ~FALSE

#define ICON_WIDTH  230
/*
     ここでウィンドウに関する定数を設定
*/
#define WDEFID      WI_STD
#define WINOPT      0
#define WINWIDTH    230
#define WINHIGHT    40
#define WINTITLE    "¥022追いかけっこゲーム"
#define EVENTMASK   EM_EVERY

#define FONTSIZE   12

#define MDEF 0x4D444546
char menudata[128] =                        /* メニューデータ*/
    "¥0¥2"                                  /* メニュー項目数-1*/
    "¥0¥0¥023このゲームについて¥0"
    "¥0¥0¥013アイコン…¥0"
     "Q¥0¥005終了¥0";


/*
     ここは定数から計算される定数
*/
#define WINOPTL      ( WINOPT & 0xf )
#define WINDEFID     ( WDEFID << 4 | WINOPTL )

/*
    ゲームに用いる定数値
*/
#define     LINEMAX 120

typedef     enum    {
    INIT,
    GAMESTART,
    SCENESTART,
    GAME,
    GAMEOVER
}   RUNMODE;

typedef struct      {
    short   c;                 /*待機時間のカウンタ*/
    short   d;                 /*いじける時間のカウンタ*/
    char    x;
    char    y;
    char    u;
    char    v;
}   MONSTER;

typedef     char    MAP[LINEMAX][50];

extern      rectImg hito;
extern      rectImg kuware;
extern      rectImg hebi;
extern      rectImg ijike;
extern      rectImg kabe;
extern      rectImg esa;
extern      rectImg power;
extern      rectImg erase;
extern      char    MapData[];

void        main(void);
int         SX_init(void);
void        initChar(void);
void        drawMap(void);
void        putScore(void);
void        newScene(void);
void        setMapBuff(void);
void        readPattern(void);
void        SX_term(void);
void        MenuPrepare(void);
void        procIDLE(void);
void        SceneStart(void);
void        game(void);
void        eaten(void);
void        gameOver(void);
void        procMSLDOWN(void);
void        toWindow(void);
void        toIcon(void);
void        procMSRDOWN(void);
void        procKEYDOWN(void);
void        procKEYUP(void);
void        procUPDATE(void);
void        procACTIVATE(void);
void        procSYSTEM(void);
void        OpenError(void);
int         MyFilter(dialog *, event *);
void        doDialog(void);
void        saveSize(void);
void        recovSize(void);
void        moveMychar(void);
void        moveMonster(void);
void        monstWait(MONSTER *);
void        monstMove(MONSTER *);
void        aka(MONSTER *);
void        pin(MONSTER *);
void        guz(MONSTER *);
short       decideU(MONSTER *);
short       decideV(MONSTER *);
short       setU(MONSTER *);
short       setV(MONSTER *);
void        putMonst(short, short, short);
void        clsMonst(short, short );
int         isEaten(void);
void        locate(int ,int);
int         rnd(int);
void        putimage(int, int, char);

typedef     menu **mhandle;


rect    updRect={0,0,WINWIDTH,WINHIGHT};

    int     hiscore;
    short   initMyX;                    /*自分の生まれてくるところ*/
    short   initMyY;

    short   initMonstX;                 /*ヘビの発生場所*/
    short   initMonstY;
/*
** グローバル変数
*/
    window  *winPtr;
    rect    winSize;
    event   eventRec;
    int activeFlag;

    int iconFlag;   /* アイコンになっているかどうか */
    int lastWhen;   /* ダブルクリックの判定用 */
    point_t oldWinSize; /* アイコン時のウィンドウサイズ記憶用 */

    int     food;                       /*エサの数*/
    int     xsize,ysize;                /*迷路の大きさ*/
    int     score,left;
    int     scene;
    RUNMODE runMode;

    int     SystemCount;
    int     GameWait;

    char    inkey;                      /*押されたキー*/

    short   myx,myy;                    /*現在、自分のいるところ  */

    MONSTER monster[4];                 /*ヘビに関するデータ*/

    MAP     map;                        /*地図*/
    MAP     mapBuffer;                  /*マップデータ*/
    mhandle mHand;

#if         0
typedef struct {
}   GVAL;

register GVAL *gp asm("a5");            /* 大域変数モドキへのポインタ */
#endif

void main(void )
{
    runMode = INIT;
    if( SX_init()==FALSE ) OpenError() ;
    while(runMode != INIT ){
TSEventAvail(EVENTMASK,(tsevent *)&eventRec);
    switch( eventRec.eWhat ){
        case E_IDLE:     procIDLE();     break;
        case E_MSLDOWN:  procMSLDOWN(); break;
        case E_MSRDOWN:  procMSRDOWN(); break;
        case E_KEYDOWN:  procKEYDOWN(); break;
```

```
		case E_KEYUP:	procKEYUP();	break;
		case E_UPDATE:	procUPDATE();	break;
		case E_ACTIVATE: procACTIVATE();break;
		case E_SYSTEM1:	procSYSTEM();	break;
		case E_SYSTEM2:	procSYSTEM();	break;
	}
	}
}


int SX_init(void)
{
	task	taskBuf;

	TSGetTdb(&taskBuf, -1);
	if((TSTakeParam(&taskBuf.command,&winSize,NULL,0,NULL,NULL)&1)==0 ){
		iconFlag=FALSE;
		*(int *)&winSize.left = TSGetWindowPos();
		setMapBuff();
		winSize.right = winSize.left + xsize * FONTSIZE + 90;
		winSize.bottom= winSize.top  + ysize * FONTSIZE + 10;
		oldWinSize.p.x= 90;
		oldWinSize.p.y= 10;
	} else{
		recovSize();
		if(winSize.top==winSize.bottom){
			iconFlag=TRUE;
		} else {
			iconFlag=FALSE;
			oldWinSize.p.x= xsize * FONTSIZE + 90;
			oldWinSize.p.y= ysize * FONTSIZE + 10;

			oldWinSize.p.x= 90;
			oldWinSize.p.y= 10;
			winSize.right = winSize.left + oldWinSize.p.x;
			winSize.bottom= winSize.top  + oldWinSize.p.y;
		}
	}
	winPtr=WMOpen(NULL, &winSize, (LASCII *)WINTITLE,TRUE,WINDEFID,
			(window *)-1,TRUE,TSGetID());
	if(winPtr == NULL) return(FALSE);
	winPtr->wOption = WINOPT;
#if	0
	map	= (MAP *)MMChHdlNew(sizeof(MAP));
	mapBuffer = (MAP *)MMChHdlNew(sizeof(MAP));
	if((int )mapBuffer <= 0) {
		DMError(0x101,"追いかけっこゲーム:メモリが足りません");
		exit(-1);
	}
#endif
	activeFlag=FALSE;
	lastWhen=-1;
		MenuPrepare();
	hiscore = 0;
	runMode = GAMESTART;
	SystemCount = 0;
	GameWait = 10;
	return( TRUE );
}


void initChar(void )
{
	short	i;

	myx = initMyX;
	myy = initMyY;
	for(i=0; i < 4; i++) {
		monster[i].c = i*10+15;
		monster[i].x = initMonstX;
		monster[i].y = initMonstY;
		monster[i].u = 0;
		monster[i].v = 0;
		monster[i].d = 0;
	}
}

void drawMap(void)
{
	char	c;
	char	buf[20];
	short	x,y;

	if(runMode < SCENESTART) return;
	GMSetGraph((graph*)winPtr);
	GMPenMode(G_BACK|G_PSET);
	GMPenMode(G_FORE|G_PSET);

	food = 0;
	for(y = 0; y < ysize; y++) {
	    for(x = 0; x < xsize; x++) {
		c = map[x][y];
		switch(c) {
		    case '.':
			food++;
		    case '*':
		    case 'O':
			putimage(x, y, c);
			break;
		    default:
			putimage(x, y, ' ');
			break;
		}
	    }
	}

	GMAPage(3);
	GMFontMode(G_PSET);
	GMFontKind(G_ROM12);
	locate(xsize+2,  0); GMDrawStrZ("SCORE");
	locate(xsize+2,  4); GMDrawStrZ("HI-SCORE");
	locate(xsize+2,  5); GMDrawStrZ("  SCORE");
	sprintf(buf, "%5d",hiscore);
	locate(xsize+2 ,  7); GMDrawStrZ(buf);
	locate(xsize+2, 9); GMDrawStrZ("SCENE");
	sprintf(buf, "%5d", scene);
	locate(xsize+2, 11); GMDrawStrZ(buf);
	GMAPage(7);
	locate(xsize+2+left, 13);GMDrawStrZ(" ");		/*前の分を消す*/
	for(x = 0; x < left; x++)
		putimage(xsize+2+x, 13, 'C');
	putScore();
}

void putScore(void)
{
	char	buf[10];

	sprintf(buf, "%5d", score);
	GMAPage(3);
	GMFontKind(G_ROM12);
	locate(xsize+2,  2);
	GMDrawStrZ(buf);
	if(hiscore < score) {
		hiscore = score;
		locate(xsize+2,  7);
		GMDrawStrZ(buf);
	}
}


void newScene(void)
{
	short	x,y;

	for(x = 0; x < xsize ; x++) {
	    for(y = 0; y < ysize ; y++) {
		map[x][y] = mapBuffer[x][y];
	    }
	}
}

void setMapBuff(void )
{
	char	c, *p;
	short	x,y;

	initMyX = 0;			/*各変数の初期化*/
	initMyY = 0;
	initMonstX = 0;
	initMonstY = 0;

	xsize = 0;
	p = MapData;
	for(y = 0; y < 50; y++) {
	    if(*p == 0 ) break;
	    for(x = 0; x < LINEMAX; x++) {
		if(*p == 0 ) break;
		c = *p++;
		if(c == '¥n') break;

		if(c == 'C') {			/*自分キャラの現れるところ*/
			initMyX = x;
			initMyY = y;
			mapBuffer[x][y] = ' ';
		} else if(c == 'M') {	/*ヘビの現れるところ*/
			initMonstX = x;
			initMonstY = y;
			mapBuffer[x][y] = ' ';
		} else {
			mapBuffer[x][y] = c;
		}
	    }
	    if(x > xsize) xsize = x;
	}

	ysize = y;

	if(initMyX == 0) initMyX = xsize / 2;
	if(initMyY == 0) initMyY = ysize / 2;

	if(initMonstX == 0) initMonstX = xsize / 2;
	if(initMonstY == 0) initMonstY = ysize / 2;
}


void SX_term(void)
{
	MMHdlDispose(mHand);
#if 0
	MMHdlDispose(map);
	MMHdlDispose(mapBuffer);
#endif
	WMDispose(winPtr);
	exit();
}


void MenuPrepare(void) {

	/* メニューハンドルの確保 */
	mHand = (mhandle) MMChHdlNew(sizeof( menu ));
	if ( mHand == (mhandle)0){
		DMError (0xff01,"メモリーが確保できません");
		exit(-1);
	}
	/* メニュー処理ルーチンの登録 */
	(**mHand).mProc = (int)RMRscGet ( (long)MDEF, 0 );

	/* アイテムの選択フラグの設定 */
	(**mHand).mEnable = 0xffffffffL;

	/* メニューアイテムの設定 */
	memcpy( &(**mHand).mData, &menudata, 128 );
}


void procIDLE(void )
{
	if(activeFlag == FALSE || iconFlag == TRUE) return;

	GMSetGraph((graph*)winPtr);
	if(SystemCount) {
		if(SystemCount > 0) SystemCount--;
		else if(++SystemCount == -1) {	/*画面にメッセージ表示中*/
			drawMap();
		} else ;
	} else {
	    SystemCount = GameWait;
	    if(runMode == GAMESTART) {
		left = 3;
		score = 0;
		scene = 1;
		newScene();
		runMode = SCENESTART;
	    }
	    else if(runMode == SCENESTART)
		SceneStart();
	    else if(runMode == GAME)	/*ゲーム中*/
		game();
	    else if(runMode == GAMEOVER)
		gameOver();
```

```c
	}
}

/*
	スタート
*/
void SceneStart(void) {
	initChar();
	drawMap();
	locate(xsize / 2 - 3, ysize / 2 - 2);
	GMAPage(3);
	GMFontKind(G_ROM24);
	GMPenMode(G_BACK|G_OR);
	GMDrawStrZ("START!!");
	SystemCount = -200;
	runMode = GAME;
}

/*
	ゲームのメインループ
*/
void game(void )
{
	moveMonster();
	if(isEaten()) eaten();
	else {
		moveMychar();
		putScore();
		if(isEaten()) eaten();
	}
	if(food == 0) {		/*面クリア*/
		scene++;
		if(GameWait) GameWait--;
		locate(xsize / 2 - 8, ysize / 2 - 2);
		GMAPage(3);
		GMFontKind(G_ROM24);
		GMPenMode(G_BACK|G_OR);
		GMDrawStrZ("CONGRATURATIONS!!");
		SystemCount = -200;
		newScene();
		runMode = SCENESTART;
	}
}

/*
	自分が食べられた時の処理
*/
void eaten(void)
{
	putimage(myx, myy, '+');
	SystemCount = -100;
	if(--left > 0)
		runMode = SCENESTART;
	else {				/* ゲームオーバー */
		GMAPage(7);
		locate(xsize+2, 13);
		GMDrawStrZ(" ");		/*残りを消す*/
		GMAPage(3);
		locate(xsize / 2 - 5, ysize / 2 - 2);
		GMFontKind(G_ROM24);
		GMPenMode(G_BACK|G_OR);
		GMDrawStrZ("GAME OVER");

		locate(xsize / 2 - 10, ysize / 2 + 5);
		GMDrawStrZ("PUSH S KEY TO RESTART");
		runMode = GAMEOVER;
		SystemCount = 0; /*画面が描き直されないようにする*/
	}
}

/*
	ゲームオーバー後「S」が押されるのを待つ
*/
void gameOver(void)
{
	SystemCount = 0;		/*画面が描き直されないようにする*/
	if(toupper(inkey) == 'S') {
		runMode = GAMESTART;
		inkey = 0;
	}
}

/*
	マウスの左ボタンが押された
*/
void procMSLDOWN(void )
{
	if((window *)eventRec.eWhom != winPtr ) return;
	if( activeFlag == FALSE ){
		WMSelect( winPtr );
		activeFlag = TRUE;
		if( EMLStill() == 0) goto checkDClick;
	}
	if(SXCallWindM(winPtr,(tsevent *)&eventRec) == W_INCLOSE) {
		SX_term();
	}

checkDClick:
	if(iconFlag==TRUE){ /* アイコンになっている */
		if(lastWhen==(-1))
			lastWhen=eventRec.eWhen;
		else{
			if((eventRec.eWhen-lastWhen)<EMDClickGet()){
				lastWhen=-1;
				toWindow();
			}
			else
				lastWhen=eventRec.eWhen;
		}
	}
	TSGetEvent(EVENTMASK,(tsevent *)&eventRec);
}

void toWindow(void)
{
	iconFlag=FALSE;
	WMSize(winPtr,oldWinSize,-1);
}

void toIcon(void)
{
	rect	r;
	point_t p;

	iconFlag=TRUE;
	r=winPtr->wGraph.grRect;
	oldWinSize.p.x=r.right-r.left;
	oldWinSize.p.y=r.bottom-r.top;
	p.p.x=ICON_WIDTH;
	p.p.y=0;
	WMSize(winPtr,p,-1);
}


void procMSRDOWN(void )
{
	int	item;

	if( winPtr != (window *)eventRec.eWhom ) return;
	TSGetEvent(EVENTMASK, (tsevent *)&eventRec);/* イベントを取除く */
	/* ウィンドウがアクティブか、OPT.1 key が押されていれば */
	/* ウィンドウ内の右クリックは、許可する。*/
	if ( ( activeFlag == TRUE ) ||
		(( EHM_OPT1 & ((event *)(&eventRec))->eHow) == EHM_OPT1))
		{
		/* メニューの選択処理を行う */
		item = MNSelect( mHand, eventRec.eWhere);
		switch(item){
			case 1:
				doDialog();
				break;
			case 2:
				if(iconFlag == TRUE)
					toWindow();
				else
					toIcon();
				break;
			case 3:
				SX_term();
				break;
		}
	}
}

void procKEYDOWN(void)
{

	if(activeFlag == FALSE) return;

	TSGetEvent(EVENTMASK, (tsevent *)&eventRec); /* イベントを取除く */

	inkey = eventRec.eWhom & 0xffff;
	if((eventRec.eHow & 0x04) && (inkey & 0x5f) == 'Q') {
		SX_term();
	}
}

void procKEYUP(void )
{
	inkey = 0;
}

void procUPDATE(void)
{
	if(runMode == INIT) return;
	if((window *)eventRec.eWhom != winPtr ) return;
	WMUpdate(winPtr);
	drawMap();
	WMUpdtOver(winPtr);
	TSGetEvent(EVENTMASK,(tsevent *)&eventRec);
}

void procACTIVATE(void)
{
	if((window *)eventRec.eWhom == winPtr )
		activeFlag = TRUE;
	else if(eventRec.eWhom && activeFlag ) {
		activeFlag = FALSE;
		TSGetEvent(EVENTMASK,(tsevent *)&eventRec);
		}
}

void procSYSTEM(void) {
	switch( ((tsevent*)&eventRec)->what2 ){
	case CLOSEALL:
	case ENDTSK:
		SX_term();
		break;
	case WINDOWSELECT:
		WMSelect( winPtr );
		break;
	case SAVE:
		saveSize();
		break;
	}
}


void OpenError(void) {
	DMError(0x101,"ウィンドウがオープンできません");
	exit(-1);
}

/*************************************
* doDialog()  ダイアログを出す関数  *
*************************************/
/*
	ここでダイアログウィンドウに関する定数を設定
*/
#define DWINDEFID	(38<<4)
#define DWINTITLE	"¥020ダイアログだよん"
/*
	アイテムリスト
*/
typedef struct dlgItem2 {
	long		dlgIHdl;
	rect		dlgIBounds;
	unsigned char	dlgIType;
	unsigned char	dlgISize;
	unsigned char	dlgIData[32];
} dlgItem2;

struct {
	short	itemNo;
	dlgItem2 dItem1;
	dlgItem2 dItem2;
	dlgItem2 dItem3;
	dlgItem2 dItem4;
	dlgItem2 dItem5;
} dItemList ={
	5-1,
	{
```


▶私の会社の同じ課に「安藤道子」という人がいたので，もしや，と思ったけど別人でした。これがきっかけで知り合いになれたからよかったですけど。　長崎　洋(23)栃木県

```
        0,
        {256-8-42,128-8-18,256-8,128-8},
        DT_STDBTN,
        32,
        "¥007 O K "
    },
    {
        0,
        {4,4,252,16},
        DT_STCTXT+DT_DISABL,
        32,
        "¥001¥024このプログラムは "
    },
    {
        0,
        {4,35,252,47},
        DT_STCTXT+DT_DISABL,
        32,
        "¥001¥022追いかけっこゲーム"
    },
    {
        0,
        {4,60,252,72},
        DT_STCTXT+DT_DISABL,
        32,
        "¥001¥031ｽｹﾙﾄﾝ：中森章 Jul.11 1991"
    },
    {
        0,
        {4,80,252,92},
        DT_STCTXT+DT_DISABL,
        32,
        "¥377¥032本体 ：石上達也 Jul.6 1992"
    }
};

/*
        ダイアログを開く位置（中央よりも少し上にしてある）
*/
rect dlBounds={ 384-128,256-64-20,384+128,256+64-20 };
/*
        フィルター関数
*/
int MyFilter(dialog *Dialog, event *ev) {
    point_t okbtn;

    if( ev->eWhat == E_KEYDOWN ){
        if((short)(ev->eWhom)==13){
            okbtn.p.x=384+128-10;
            okbtn.p.y=256+64-20-10;
            ev->eWhere=okbtn;
            ev->eWhat =E_MSLDOWN;
        }
    }
    return 0;
}

void doDialog(void)
{

     dialog   *dialogPtr;
     dlgIList **dIHdl;
     int  ditem;

     dIHdl=(dlgIList**)MMChHdlNew( sizeof(dItemList) );
    .if( dIHdl == NULL ){
    DMError(0x101,"領域確保に失敗しました。");
    return;
     }
     memcpy(*dIHdl,&dItemList,sizeof(dItemList));
     dialogPtr=DMOpen(NULL,&dlBounds,(LASCII *)DWINTITLE,TRUE,DWINDEFID,
            (window *)-1,TRUE,TSGetID(),dIHdl);
     if( dialogPtr == NULL ){
    MMHdlDispose(dIHdl);
    DMError(0x101,"ウインドウがオープンできません。");
    return;
     }
     DMBeep(2);
     ditem=DMControl((void*)MyFilter );
     DMDispose(dialogPtr);
     MMHdlDispose(dIHdl);

}

void saveSize(void )
{
     task    taskBuf;
     int len;
     int i;
     char    BUF[256];

     sprintf(BUF,"-S%d,%d ",oldWinSize.p.x,oldWinSize.p.y);
     len=strlen(BUF);
     if(len>255) len=255;
     TSGetTdb(&taskBuf, -1);
     for(i=0;i<len;i++)
         taskBuf.command.Lstr[i]=BUF[i];
     taskBuf.command.length=len;
     TSSetTdb(&taskBuf, -1);
}

void recovSize(void )
{
     task    taskBuf;
     int x,y;
     char    BUF[256];

     TSGetTdb(&taskBuf, -1);
     sscanf(taskBuf.command.Lstr,"-S%d,%d %s",&x,&y,BUF);
     oldWinSize.p.x=x;
     oldWinSize.p.y=y;
}

/*
     自分を動かす
*/
void moveMychar(void )
{
char        c;
short       i;
short oldx,oldy;

#ifndef KEYUP_EVENT /*キーアップイベントの取り方がわかったら消去する*/
    char     kg8,kg9;

    kg8 = BITSNS(8);
    kg9 = BITSNS(9);

    if(kg8 & 0x10) inkey = '8';
    else if(kg8 & 0x80) inkey = '4';
    else if(kg9 & 0x02) inkey = '6';
    else if(kg9 & 0x10) inkey = '2';
    else inkey = 0;

#endif

    oldx = myx;
    oldy = myy;

    switch(inkey) {
        case '2':
            if(myy < ysize) myy++;
            break;
        case '4':
            if(myx == 0) myx = xsize-1;      /*ワープ*/
            else     myx--;
            break;
        case '6':
            if(myx == xsize-1) myx = 0;      /*ワープ*/
            else     myx++;
            break;
        case '8':
            if(myy > 0) myy--;
            break;
    }

    c = map[myx][myy];
    if(c == '*') {
        myx = oldx;
        myy = oldy;
    }
    GMFontMode(G_PSET);
    GMFontKind(G_ROM12);
    if(myx != oldx || myy != oldy) {
        if(c == '.') {
            food--;
            map[myx][myy] = ' ';
            score += 10;
        } else if(c == 'O') {
            map[myx][myy] = ' ';
            score += 50;
            for(i = 0; i < 4 ; i++) {
                monster[i].d = 600;
            }
        }
        putimage(oldx, oldy, ' ');
    }
    putimage(myx, myy, 'C');
}

/*
    ヘビを動かす
*/
void moveMonster(void ) {
short     i;
    for(i = 0; i < 4 ; i++) {
        if(monster[i].c >0) monstWait(&monster[i]);
        else                monstMove(&monster[i]);
    }
}

void monstWait(MONSTER *p) {
    clsMonst(p->x, p->y);

    if(--p->c > 0) {
        p->x = initMonstX;
        p->y = initMonstY;
    } else {
        p->x = initMonstX;
        p->y = initMonstY;
        p->u = 0;
        p->v = -1;
        p->c = 0;
    }

    putMonst(p->x, p->y, p->d);
}


void monstMove(MONSTER * p) {
short     hk = 0;
short     oldx, oldy;
short     x,y,u,v;

    oldx = x = p->x;
    oldy = y = p->y;
    u = p->u;
    v = p->v;

    if(x > 0           && map[x-1][y] != '*') hk++;
    if(x < xsize-1 && map[x+1][y] != '*') hk++;
    if(y > 0           && map[x][y-1] != '*') hk++;
    if(y < ysize-1 && map[x][y+1] != '*') hk++;

    if(hk > 2) {      /*曲がれる方向が2か所以上ある*/
        switch(rand() & 3) {
            case 0:
                aka(p);
                break;
            case 1:
                pin(p);
                break;
            default:
                guz(p);
                break;
        }
    }

    u = p->u;
    v = p->v;

    while(map[x+u][y+v] == '*') {
        if(v == 0) {
            u = 0;
            v = decideV(p);
            if(map[x][y+v] == '*') v = -v;
        } else {
            u = decideU(p);
            v = 0;
            if(map[x+u][y] == '*') u = -u;
        }
    }
```

▶Oh!X11月号171ページの広告。X68000 Compact XVIが倒れているのがかわいそうだ。
丸山　淳平(18)神奈川県

```
    x += u;
    y += v;

    if(x == xsize-1) x = 1;
    else if(x == 0) x = xsize-2;
    if(p->d > 0) p->d--;

    p->x = x;
    p->y = y;
    p->u = u;
    p->v = v;

    clsMonst(oldx, oldy);
    putMonst(x, y, p->d);
}

void aka(MONSTER *p) {
short       u;
    u = setU(p);
    if(u == 0 || map[p->x+u][p->y] == '*') {
            p->u = 0;
            p->v = decideV(p);
    } else {
            p->u = u;
            p->v = 0;
    }
}

void pin(MONSTER *p) {
short       v;
    v = setV(p);
    if(v == 0 || map[p->x][p->y+v] == '*') {
            p->u = decideU(p);
            p->v = 0;
    } else {
            p->u = 0;
            p->v = v;
    }
}

void guz(MONSTER *p) {
short u,v;
    if(p->u == 0) {
            u = (rand() & 1)*2-1;
            if(map[p->x+u][p->y] == '*') {
                    p->u = 0;
                    p->v = decideV(p);
            } else {
                    p->u = u;
                    p->v = 0;
            }
    } else {
            v = (rand() & 1)*2-1;
            if(map[p->x][p->y+v] == '*') {
                    p->u = decideU(p);
                    p->v = 0;
            } else {
                    p->u = 0;
                    p->v = v;
            }
    }
}

short decideU(MONSTER *p) {
short       u;
    u = setU(p);
    if(u == 0) {
            u = (rand() & 1)*2 - 1;
    }
    return(u);
}

short decideV(MONSTER *p) {
short       v;
    v = setV(p);
    if(v == 0) {
            v = (rand() & 1)*2 - 1;
    }
    return(v);
}

short setU(MONSTER *p) {
short       u;

    if(myx == p->x) u = 0;
    if(myx >  p->x) u = 1;
    if(myx <  p->x) u =-1;
    if(p->d) u=-u;

    if(p->u == -u) u = - u;
    return(u);
}

short setV(MONSTER *p) {
short       v;

    if(myy == p->y) v = 0;
    if(myy >  p->y) v = 1;
    if(myy <  p->y) v =-1;
    if(p->d) v=-v;

    if(p->v == -v) v = - v;
    return(v);
}

void putMonst(short x, short y, short k) {
    if(k)    putimage(x, y, 'a');
    else     putimage(x, y, 'A');
}

void clsMonst(short x, short y) {

    switch(map[x][y]) {
        case '*':
            putimage(x, y, '*');
            break;
        case '.':
            putimage(x, y, '.');
            break;
        case 'O':
            putimage(x, y, 'O');
            break;
        default:
            putimage(x, y, ' ');
            break;
    }
}

int isEaten() {
short       i;
    for(i = 0; i < 4 ; i++) {
        if(monster[i].x == myx && monster[i].y == myy) {
            if(monster[i].d == 0) {
                    return(1);
            } else {          /*いじけている時(ヘビが食べられた)*/
                    score += 100;
                    monster[i].c = 15;
                    monster[i].x = initMonstX;
                    monster[i].y = initMonstY;
                    monster[i].u = 0;
                    monster[i].v = 0;
                    monster[i].d = 0;
            }
        }
    }
    return(0);
}


void locate(int x, int y) {
    point_t loc;

    loc.p.x = FONTSIZE * x + 6;
    loc.p.y = FONTSIZE * y + 6;
    GMMove(loc);
}

int rnd(int i) {
    return(rand() % i);
}


void putimage(int x, int y, char c) {
point_t             loc;

    loc.p.x = x * FONTSIZE + 6;
    loc.p.y = y * FONTSIZE + 6;

    GMSetGraph((graph *)winPtr);
    GMAPage(15);
    switch(c) {
        case 'C':
            GMPutRImg((rectImg *)&hito, loc);
            break;
        case '+':
            GMPutRImg((rectImg *)&kuware, loc);
            break;
        case 'A':
            GMPutRImg((rectImg *)&hebi, loc);
            break;
        case 'a':
            GMPutRImg((rectImg *)&ijike, loc);
            break;
        case '.':
            GMPutRImg((rectImg *)&esa, loc);
            break;
        case 'O':
            GMPutRImg((rectImg *)&power, loc);
            break;
        case '*':
            GMPutRImg((rectImg *)&kabe, loc);
            break;
        default:
            GMPutRImg((rectImg *)&erase,loc);
            break;
    }
}
```

リスト3

```
/*
    SXおいかけっこゲーム
    パターン作成プログラム
    Programmed By T.Ishigami
    Jul 12th 92
*/

#include   <stdio.h>

/*
 * レクタングルレコード (rectangle)
 */
typedef struct rect {
    short   left;                       /* 左端の座標           */
    short   top;                        /* 上端の座標           */
    short   right;                      /* 右端の座標           */
    short   bottom;                     /* 下端の座標           */
} rect;

typedef struct      {         /*12*12 のレクタングルイメージ*/
    rect    bounds;
    char    pat[96];
} RectImg;

#define     PATNUM  8                   /*キャラクタパターンの数*/
char        *patName[] = {
    "hito",
    "ijike",
    "hebi",
    "kabe",
    "kuware",
    "power",
    "erase",
    "esa"
};

FILE        *fd;

main()
{
    int     i;

```

▶サンポールは死ぬほどまずいです。　大島　大介(16)北海道

```
    fd = fopen("packpat.s","w");
    if(fd == NULL) {
            OpenError("packpat.s");
    }
    fprintf(fd, "*おいかけっこゲームで使うキャラクタパターンです¥n");
    fprintf(fd, "*ASでアセンブルの上、本体とリンクして下さい¥n");
    fprintf(fd, "¥t.data¥n");
    fprintf(fd, "¥t.even¥n");
    for(i=0; i < PATNUM; i++) {
            fprintf(fd, "¥t.global¥t_%s¥n", patName[i]);
    }
    fprintf(fd, "¥t.global¥t_MapData¥n");
    mapWrite();
    for(i=0; i < PATNUM; i++) {
            fileWrite(patName[i]);
    }
    fprintf(fd, "¥t.end¥n");
    fclose(fd);
}

mapWrite()
{
    FILE    *fd1;
    char    buf[80];
    int     i;

    fd1 = fopen("map.dat","r");
    if(fd1 == NULL) {
            OpenError("map.dat");
    }
    fprintf(fd, "_MapData:¥n");
    while(fgets(buf, 80, fd1)) {
            for(i = 0; i < 80 ; i++) {       /*改行をとる*/
                    if(buf[i] == '¥n') {
                            buf[i] = 0;
                            break;
                    }
            }
            fprintf(fd, "¥t.dc.b¥t¥'%s¥',10¥n",buf);
    }
    fclose(fd1);
}

fileWrite(char *patName)
{
    char    fileName[22];
    FILE    *fd1;
    RectImg pat;
    int     i;

    strncpy(fileName, patName, 19);
    strcat(fileName, ".pat");
    fd1 = fopen(fileName,"rb");
    if(fd1  == NULL) {
            OpenError(fileName);
    }

    fread((void *)&pat, 1, sizeof(RectImg), fd1);
    fclose(fd1);

    fprintf(fd, "_%s:¥n",patName);                  /*パターン名*/
    fprintf(fd, "¥t.dc.w¥t%d,%d,%d,%d¥n",    /*レクタングル*/
            pat.bounds.left,
            pat.bounds.top,
            pat.bounds.right,
            pat.bounds.bottom);
    for(i=0; i < 96; i++) {                          /*イメージ本体*/
            if((i & 7) == 0) fprintf(fd, "¥t.dc.b¥t");
            if((pat.pat[i]& 0xff) < 0x10)
                    fprintf(fd, "$0%x", pat.pat[i] & 0xff);
            else
                    fprintf(fd, "$%x",  pat.pat[i] & 0xff);

            if((i & 7) == 7) fprintf(fd, "¥n");
            else             fprintf(fd, ",");
    }
}

OpenError(char *fileName)
{
    printf("%sがアクセス出来ません¥n",fileName);
    exit(-1);
}
```

リスト4

```
*おいかけっこゲームで使うキャラクタパターンです
*ASでアセンブルの上、本体とリンクして下さい
    .data
    .even
    .global _hito
    .global _ijike
    .global _hebi
    .global _kabe
    .global _kuware
    .global _power
    .global _erase
    .global _esa
    .global _MapData
_MapData:
    .dc.b   '*****************************',10
    .dc.b   '*...........................*',10
    .dc.b   '*O*************************O*',10
    .dc.b   '*.*************************.*',10
    .dc.b   '*...........................*',10
    .dc.b   '*****.*****************.*****',10
    .dc.b   '    *.*****************.*',10
    .dc.b   '*****.........M.........*****',10
    .dc.b   '     .*****************.',10
    .dc.b   '*****...................*****',10
    .dc.b   '    *.*****************.*',10
    .dc.b   '*****.*****************.*****',10
    .dc.b   '*.............C.............*',10
    .dc.b   '*.*************************.*',10
    .dc.b   '*O*************************O*',10
    .dc.b   '*...........................*',10
    .dc.b   '*****************************',10
_hito:
    .dc.w   0,0,12,12
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $0e,$00,$11,$00,$11,$00,$11,$00
    .dc.b   $0e,$00,$04,$00,$7f,$c0,$04,$00
    .dc.b   $04,$00,$0a,$00,$11,$00,$20,$80
    .dc.b   $0e,$0f,$11,$0f,$11,$0f,$11,$0f
    .dc.b   $0e,$0f,$04,$0f,$7f,$cf,$04,$0f
    .dc.b   $04,$0f,$0a,$0f,$11,$0f,$20,$8f
_ijike:
    .dc.w   0,0,12,12
    .dc.b   $00,$00,$00,$00,$60,$00,$f7,$00
    .dc.b   $f5,$00,$35,$00,$15,$00,$15,$00
    .dc.b   $15,$20,$15,$20,$08,$c0,$00,$00
    .dc.b   $00,$00,$00,$00,$60,$00,$f7,$00
    .dc.b   $f5,$00,$35,$00,$15,$00,$15,$00
    .dc.b   $15,$20,$15,$20,$08,$c0,$00,$00
    .dc.b   $00,$00,$00,$00,$60,$00,$f7,$00
    .dc.b   $f5,$00,$35,$00,$15,$00,$15,$00
    .dc.b   $15,$20,$15,$20,$08,$c0,$00,$00
    .dc.b   $00,$0f,$00,$0f,$60,$0f,$f7,$0f
    .dc.b   $f5,$0f,$35,$0f,$15,$0f,$15,$0f
    .dc.b   $15,$2f,$15,$2f,$08,$cf,$00,$0f
_hebi:
    .dc.w   0,0,12,12
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$60,$00,$f7,$00
    .dc.b   $f5,$00,$35,$00,$15,$00,$15,$00
    .dc.b   $15,$20,$15,$20,$08,$c0,$00,$00
    .dc.b   $00,$00,$00,$00,$60,$00,$f7,$00
    .dc.b   $f5,$00,$35,$00,$15,$00,$15,$00
    .dc.b   $15,$20,$15,$20,$08,$c0,$00,$00
    .dc.b   $00,$0f,$00,$0f,$60,$0f,$f7,$0f
    .dc.b   $f5,$0f,$35,$0f,$15,$0f,$15,$0f
    .dc.b   $15,$2f,$15,$2f,$08,$cf,$00,$0f
_kabe:
    .dc.w   0,0,12,12
    .dc.b   $00,$00,$00,$00,$3f,$c0,$3f,$c0
    .dc.b   $30,$c0,$30,$c0,$30,$c0,$30,$c0
    .dc.b   $3f,$c0,$3f,$c0,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$3f,$c0,$3f,$c0
    .dc.b   $30,$c0,$30,$c0,$30,$c0,$30,$c0
    .dc.b   $3f,$c0,$3f,$c0,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $ff,$ff,$ff,$ff,$ff,$ff,$ff,$ff
    .dc.b   $ff,$ff,$ff,$ff,$ff,$ff,$ff,$ff
    .dc.b   $ff,$ff,$ff,$ff,$ff,$ff,$ff,$ff
_kuware:
    .dc.w   0,0,12,12
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$40,$40,$20,$80,$11,$00
    .dc.b   $0a,$00,$04,$00,$0a,$00,$11,$00
    .dc.b   $20,$80,$40,$40,$00,$00,$00,$00
    .dc.b   $00,$0f,$40,$4f,$20,$8f,$11,$0f
    .dc.b   $0a,$0f,$04,$0f,$0a,$0f,$11,$0f
    .dc.b   $20,$8f,$40,$4f,$00,$0f,$00,$0f
_power:
    .dc.w   0,0,12,12
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$0f,$1f,$8f,$30,$cf,$60,$6f
    .dc.b   $40,$2f,$40,$2f,$40,$2f,$40,$2f
    .dc.b   $60,$6f,$30,$cf,$1f,$8f,$00,$0f
_erase:
    .dc.w   0,0,12,12
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$0f,$00,$0f,$00,$0f,$00,$0f
    .dc.b   $00,$0f,$00,$0f,$00,$0f,$00,$0f
    .dc.b   $00,$0f,$00,$0f,$00,$0f,$00,$0f
_esa:
    .dc.w   0,0,12,12
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$00,$00,$00,$00,$00,$00,$00
    .dc.b   $00,$0f,$00,$0f,$00,$0f,$06,$0f
    .dc.b   $0f,$0f,$1f,$8f,$1f,$8f,$0f,$0f
    .dc.b   $06,$0f,$00,$0f,$00,$0f,$00,$0f
    .end
```

▶ DōGAのCGAシステムにはまりました。X68000使用時間の90%はCGAにかけている現状です。CADは面倒なので，もっと簡単な地形のサーフェスデータ作成ツールを自作したり，黒を透明色にするフィルタを作ったり……たいへん！　　鈴木　浩之(20)宮城県

リスト5

(9967バイト，圧縮ファイル)

```
0000  1F BD 2D 6C 68 35 2D CD  : 0C
0008  26 00 00 DA 4B 00 00 05  : 50
0010  05 F0 18 20 01 06 70 61  : 05
0018  63 6B 2E 78 6F 0D 48 00  : 38
0020  00 1F CC 8C DD F7 66 DA  : 8B
0028  72 77 FB F2 4D 35 3D 8A  : 1F
0030  28 9A 88 63 14 63 8E 7B  : 2D
0038  13 71 8F 5C 91 A5 1C 63  : 24
0040  1C 95 90 49 24 E0 9A 1B  : 43
0048  49 09 37 06 32 53 DF 53  : 46
0050  69 7B EB 72 36 D2 89 B6  : 88
0058  C7 1D 74 13 4E 42 B0 A5  : 50
0060  24 B2 16 6C E9 87 5C 96  : BA
0068  1B 87 16 53 4B 09 A5 38  : 3C
0070  2E 4D B8 E4 25 94 0E 29  : 07
0078  5D 94 B2 94 E3 82 53 82  : 71
-------------------------------------
SUM:  B9 09 0D 26 08 69 46 B7  F94A

0080  43 66 96 12 98 59 B2 CA  : BE
0088  C6 BE FD FD FB D4 DC 61  : 8A
0090  40 02 97 FE EE 6E ED CD  : ED
0098  B9 FF 3B 3B C0 EC EF 0B  : D4
00A0  73 BB 3B C1 EF 27 BC 0E  : 0A
00A8  F0 3B B3 3B E0 23 BC 6E  : 46
00B0  F8 6C BD DD A4 6D C6 E4  : B9
00B8  AF 2C 88 A7 EE 9D 69 09  : 07
00C0  D4 F8 C4 22 DF 04 84 AB  : C4
00C8  ED C8 45 8F 08 45 DC C2  : 74
00D0  7F 8F E9 04 A2 E1 10 9D  : 2B
00D8  5E 33 EF 1C 2E 71 1E C9  : 22
00E0  48 84 DB 8D B6 18 28 90  : BA
00E8  98 AF 36 40 2F 88 D6 B0  : FA
00F0  84 97 2C 42 9F D9 C0 7E  : 3F
00F8  DB 52 20 DE 63 48 CE EF  : 93
-------------------------------------
SUM:  E9 51 D6 86 40 37 2B EC  D377

0100  F0 ED 62 63 C9 A8 F9 87  : 93
0108  2D 7D FA D8 AB 25 01 28  : 75
0110  C9 68 4D 28 7D A9 25 61  : 52
0118  1D 85 75 E1 F6 D0 96 24  : 78
0120  B1 13 BC D7 96 8D 50 D7  : A1
0128  9C A8 46 9C 6B 5F AF 46  : E5
0130  90 B5 E4 A3 42 35 E7 75  : 9F
0138  88 CF 16 B7 C8 B0 2D 7B  : 44
0140  B4 56 96 BE 43 DF 1C 71  : 0D
0148  0A 44 97 21 F5 F3 2D BF  : DA
0150  DE 95 BD 95 F5 E5 7D 99  : B5
0158  5C 29 5E 1E 27 5F 87 D9  : E7
0160  62 49 23 ED 6E 49 1A E7  : 73
0168  1E 60 4B 97 D6 F3 06 C9  : F8
0170  71 4F 1D E6 01 09 EE FD  : B8
0178  1C 97 C7 79 B0 25 A1 47  : B0
-------------------------------------
SUM:  6D 7D B4 86 3B 97 C4 D7  A898

0180  F2 5F FB 24 3C E0 CE 89  : E3
0188  5A A8 C0 FD 21 0F A2 5D  : EE
0190  D6 FE A1 27 A4 92 37 66  : 6F
0198  FD 43 C7 36 FA 98 F2 71  : 32
01A0  C7 01 FA 9E B4 9C 78 58  : 80
01A8  F8 44 B4 EA 9B 91 61 44  : AB
01B0  F2 54 13 C9 01 3B 02 2B  : 8B
01B8  01 6B 80 5A D7 02 83 E5  : 87
01C0  B6 C8 25 A3 6E 09 17 23  : F7
01C8  DA 8A 2D 8D D9 42 F9 F5  : 27
01D0  14 A4 59 C0 80 92 E9 12  : DE
01D8  CC B5 45 19 8B F3 1A FC  : 73
01E0  0B 10 57 B6 92 EB D7 22  : 9E
01E8  BF 81 62 0A DA 89 20 A1  : D0
01F0  B4 A8 05 2D 69 71 BD 43  : 68
01F8  2E 34 0D 40 A3 7D 0A 8A  : 63
-------------------------------------
SUM:  ED 64 1F 5F EC B5 C8 1F  DA94

0200  62 2B E3 75 FD 4B 82 15  : C4
0208  A2 E8 98 35 20 B2 13 7D  : B9
0210  56 6B 78 39 3C 05 18 A8  : 73
0218  31 F2 92 B2 A2 C6 03 BE  : 90
0220  60 B1 B4 96 47 22 2E 4F  : 41
0228  11 FC C7 36 E4 D6 29 89  : 76
0230  7D A5 31 BF 38 AF 4E CD  : 14
0238  40 BD 09 0A 8D 08 EA 5F  : EE
0240  C9 90 B8 6B D5 AB 8F 2E  : B9
0248  92 EF DC C8 7D D9 BE BE  : F7
0250  D4 06 1F D5 84 51 EA CE  : 5B
0258  2D 95 C1 F5 7D E9 07 39  : 1E
0260  EA D8 36 56 F3 1B 95 26  : 17
0268  A1 5F 91 90 45 DC 6C 22  : D0
0270  4E 69 F0 CE 26 78 13 FA  : 20
0278  B6 A7 2C 2D 7A 73 A6 DC  : 25
-------------------------------------
SUM:  A4 E0 91 08 16 17 37 0D  0D1B

0280  B6 39 68 7D CD 39 8E 72  : DA
0288  B8 2E 7C 4C 22 4D 46 F2  : 55
0290  5F 5F 42 74 FE 0D 69 1E  : 06
0298  57 69 8D 60 F3 CD 5E 8F  : 5A
02A0  A9 88 48 4E AB 5F 56 93  : BA
02A8  0D B8 40 FC BF 61 E7 2A  : 32
02B0  82 F5 57 88 58 49 B9  : 32
02B8  22 AE 6B 51 69 7C C2 E7  : 1A
02C0  3D 5B CC 05 44 17 EB A2  : 51
02C8  78 FB 70 56 13 B8 BD 25  : E6
02D0  17 0C 16 B5 18 04 06 12  : 22
02D8  BD B3 BF 0C 5A DF 26 E7  : 81
02E0  B5 A6 09 09 D5 D5 5F 91  : 07
02E8  B3 A2 81 F5 2E 51 AA 76  : 6A
02F0  DA 69 E6 33 42 71 B0 F8  : B7
02F8  D2 79 41 F7 8D 86 7D 6F  : 82
-------------------------------------
SUM:  1B DE 5D D3 D6 C3 ED 9C  DD08

0300  E1 E1 31 D4 62 93 7F 58  : 93
0308  4A 74 D9 5D 9C 15 D3 26  : 9E
0310  C5 19 E0 AD FA 67 22 C2  : B0
0318  08 E4 FA 87 92 5E 63 35  : F5
0320  22 2B 01 5E 7B 5C E5 43  : AB
0328  A6 B9 B1 C1 79 FE 00 AB  : F3
0330  E0 BA 98 9F BA DE C1 77  : A1
0338  31 7B 75 8A 05 73 55 C3  : 3B
0340  89 E0 BE EF 32 9D 8E BB  : 2E
0348  2A CA 43 66 8C B0 7B 63  : B7
0350  29 AA B1 CA 39 FD 70 65  : 59
0358  1C 1C 9F 47 AE 13 58 EF  : 26
0360  BE A0 10 AC CB 21 7D B8  : 3B
0368  17 CE 03 F8 65 13 37 98  : 27
0370  24 02 FD A7 DF 50 AC 97  : 3C
0378  A0 16 FC 3B FF 63 8F DD  : BB
-------------------------------------
SUM:  62 61 00 99 F0 5C 92 D3  3447

0380  B5 9A 2A 9F 04 0A F3 5F  : 78
0388  B1 0B 46 02 C5 07 9B FD  : 68
0390  88 3F 9E 64 1D 97 7E 16  : 11
0398  EE BA 27 F9 CF 94 FF 34  : 5E
03A0  5C B3 9D 18 E9 B2 DF 58  : 96
03A8  A9 20 6D 48 54 5F 54 FB  : 80
03B0  EA 5F 8B 8F 9D 44 F0 23  : 57
03B8  74 24 ED 9D 58 BC FC 25  : 57
03C0  65 2D 4E F8 59 FD 96 FA  : BE
03C8  EB 8E 21 56 73 6B F9 95  : 5C
03D0  75 72 FC E1 88 50 80 BE  : DA
03D8  73 CE 53 16 65 C2 BE 13  : A2
03E0  CC 1F 73 F2 5F F3 BD DC  : 3B
03E8  B2 6B 30 7D D6 AA AD A1  : 98
03F0  B2 AE 1B 55 55 C6 6A BD  : 12
03F8  E3 59 AE B4 CD 4C AD 0C  : 70
-------------------------------------
SUM:  8A 80 E1 47 F7 76 78 E7  A0DC

0400  A3 FA 1C 0F 1C 52 77 34  : E1
0408  34 42 BB F0 8A CE 43 28  : E4
0410  E5 9D 89 AB ED C5 7F 31  : 18
0418  99 D2 F9 63 82 DE 72 99  : 32
0420  B1 BD 23 67 2A 42 5C 0A  : CA
0428  52 C2 ED 20 53 65 C0 A8  : 41
0430  C2 21 1B 57 29 0E F4 D5  : 55
0438  0A DE 70 8F 54 13 F3 87  : C8
0440  CA 0F B8 F2 69 1D FE 4C  : 53
0448  6A 37 5F 8F 1B A6 E0 6D  : 9D
0450  89 B6 24 1A E9 85 37 5C  : 7E
0458  C8 2E 66 23 EE B8 35 FE  : 58
0460  99 CC C5 4B 42 47 3B 21  : 5A
0468  97 B2 D2 B8 B4 D3 96 96  : 86
0470  D8 44 A2 E2 89 5B A0 31  : 55
0478  EA 9B 60 12 98 AB B7 B9  : AA
-------------------------------------
SUM:  9B B0 2E 2F 81 AB 20 E8  ACBC

0480  E4 C8 4E EE FC 95 2E C2  : 69
0488  CE BE 77 3F E2 56 11 72  : FD
0490  76 A1 4B A4 98 6A 3C 81  : C5
0498  44 EA 38 82 89 45 90 03  : 49
04A0  B0 4A 94 A0 E9 CA 0D B7  : A5
04A8  C7 21 BD 2F 04 87 E2 91  : D2
04B0  16 7F 20 B9 58 A4 82 EB  : D7
04B8  C5 C2 27 69 1B 6F 32 3D  : 10
04C0  63 2D A2 F2 46 B8 2E FC  : 4C
04C8  6C 22 69 E3 12 5C 77 9C  : 5B
04D0  4A 2E 20 E5 F7 A5 C8 E9  : CA
04D8  32 36 A4 77 2A E0 C9 F5  : 4B
04E0  24 EA 31 84 19 44 AE 50  : 1E
04E8  4F 27 A2 25 71 44 D3 FC  : C1
04F0  46 74 E5 12 DB E2 33 6D  : 0E
04F8  5F E9 93 01 F5 F0 C1 1D  : 9F
-------------------------------------
SUM:  21 DE FA 31 32 F1 59 74  A908

0500  D3 42 C0 2F 83 D1 D0 BF  : E7
0508  C6 1A 74 1C F8 0F DF 5F  : B5
0510  62 61 E2 4A C4 EE 70 FB  : 0C
0518  29 5B EE E6 EB EA 7F B6  : 62
0520  57 DD 95 F6 25 7A 32 BC  : 4C
0528  79 5F A0 76 71 54 5D B3  : C3
0530  74 5C 31 D5 46 E9 A5 9B  : 45
0538  F4 A8 5E 60 CB C7 2E E9  : 03
0540  6A 8A 27 90 3A C1 99 16  : 55
0548  31 6A 30 E2 D6 B2 C1 DD  : D3
0550  6E B0 A3 79 83 AB CD D3  : 08
0558  46 90 28 83 64 F6 90 03  : 6E
0560  57 57 04 5C 04 72 47 5B  : 26
0568  1E A7 C6 90 CB 8D B6 6A  : 93
0570  54 3B 49 F1 91 37 D7 C5  : 2D
0578  DC 97 DA 07 73 E3 48 96  : 88
-------------------------------------
SUM:  50 5C D7 6E 9B 63 D3 AB  770F

0580  31 13 D5 FA 41 6F 33 1B  : 11
0588  EA EC DF 94 8F 25 3A 38  : 6F
0590  CB A1 16 C8 6F C7 DC C1  : 1D
0598  2D 95 0C F2 52 C0 14 9A  : 80
05A0  BF 55 18 E3 18 BA B5 65  : FB
05A8  16 AB 80 6D E0 69 64 4D  : A8
05B0  0E BA B6 5B E3 6B CB 65  : 57
05B8  E1 1B 5F 5B 2E FC BB 95  : 30
05C0  E1 48 56 6D 80 D1 A9 90  : 76
05C8  AD 43 12 CC A9 DD E2 2D  : 63
05D0  24 4D 66 01 CD D0 22 C6  : 5D
05D8  42 C5 00 97 69 FC E1 03  : E7
05E0  5C BB 17 10 6C 74 5A 22  : 9A
05E8  CB BA BD 79 B0 3E 63 9B  : A7
05F0  3D BB CB 79 79 39 8F 1C  : 99
05F8  A0 7D C2 CF CE BB DD 9B  : AF
-------------------------------------
SUM:  CF 54 B2 F0 5C C5 B3 54  0122

0600  B5 73 26 ED 51 AC BC 06  : FA
0608  F7 BD 2D 98 F4 59 12 E5  : BD
0610  CB AB 2E 57 9A 36 2F 06  : 00
0618  7E 12 4E E9 FB EE B6 0B  : 71
0620  8B 1B 20 FB 5C FD F4 25  : 33
0628  30 3B 2E 40 D2 CF 92 1C  : 28
0630  46 4A F3 D8 F8 86 D1 46  : F0
0638  60 54 CB 8F 73 7C E7 A5  : 89
0640  D7 54 38 9E EC C5 A7 D9  : 32
0648  77 E5 BE AD 35 B0 8B 5F  : 96
0650  F0 C8 BC FF F8 64 5B 4F  : 79
0658  7F 22 FB CD EC D9 2F AD  : 0A
0660  71 61 5A 7B FD 6D 6C 12  : 8F
0668  40 B2 84 F0 F8 47 F5 D5  : 6F
0670  BD 89 F6 2F CF 83 5A 56  : 6D
0678  C6 5E 25 A4 D2 36 18 FF  : 0C
-------------------------------------
SUM:  47 FE 81 BC 0E 16 80 98  36A5

0680  5E D4 E2 D6 74 CD 50 64  : DF
0688  77 A6 DF 04 71 CD 96 F8  : CC
0690  62 65 E9 6C C6 5F 94 F7  : CC
0698  C8 4E 28 E9 B3 D6 9B 68  : B3
06A0  B1 DC CF 3B D1 D3 94 D9  : A8
06A8  1F 07 9F F4 0D 8A 4B 94  : 2F
06B0  77 1F C1 80 FE BC A6 02  : 39
06B8  A6 77 2E D3 B9 99 EE 98  : F6
06C0  C9 1F 7D 16 49 BF 81 17  : 1B
06C8  CE 99 81 65 9B 5C 58 F2  : 8E
06D0  16 7B DC 27 C6 4C E9 10  : 9F
06D8  7D 7D 32 FB 11 9E BA 43  : D3
06E0  6F 2A 59 55 D6 D9 BF 91  : 46
06E8  0B 81 25 26 13 38 6C B9  : 47
06F0  29 08 68 2B 34 5C D7 C5  : F0
06F8  2C 5E F3 01 F8 EF 0D 2F  : A1
-------------------------------------
SUM:  E5 67 14 F5 C3 E2 13 5C  B380

0700  29 1F 97 2D 6D 2E 54 E2  : DD
0708  4B ED 16 ED F8 4B CC 94  : DE
0710  CF 53 9A 6E 59 2E 0C 07  : C4
0718  F2 33 40 A3 2C 21 E6 A8  : E3
0720  26 4D BA 55 65 AF 9C 9D  : CF
0728  25 20 A4 7C BD CF 8F 33  : B3
0730  8B 7B 4C 5D FE 25 B3 1C  : A1
0738  CE 30 39 CE 4A 2B 86 6A  : 6A
0740  7A BF 2C 40 FB FC EB DA  : 61
0748  77 50 1F D2 68 72 B3 2F  : 74
0750  A4 3A 38 95 7F CB E8 E0  : BD
0758  5D D8 BE 32 ED 1A 4C 09  : 81
0760  65 D5 9F D2 37 1F 53 C3  : 17
0768  95 4B 4A 52 1C E8 BC A5  : E1
0770  E9 55 3C 76 F3 89 46 B2  : 64
0778  94 83 5D 22 CE F1 D5 47  : 71
-------------------------------------
SUM:  42 C3 2D BC 37 6A 72 CE  0B48

0780  0E AD 56 E9 BD 46 CF 1B  : E7
0788  82 2A F9 3C 12 72 48 47  : F4
0790  2D CA A3 21 A1 34 C9 B0  : 09
0798  D5 34 22 4D AE 70 BC D6  : 28
07A0  78 A5 67 BC CE 06 E1 2E  : 23
07A8  D7 E3 C5 AA 30 3F 69 35  : 36
```

▶会社に１カ所だけウォシュレットがあります。なかなか気持ちいいので，人目を忍んで通っています。

中村　健(22)埼玉県

```
07B0  F1 D6 63 7D 82 2F 0D 09  : 6E
07B8  64 A5 62 5C 5B E6 E5 EB  : D8
07C0  1E D3 C2 02 95 7C 09 29  : F8
07C8  CF DD 1C 1A 4E C9 4D 72  : B8
07D0  2D 83 46 19 29 D1 A3 4A  : F6
07D8  23 E7 19 22 7E 53 4D 39  : 9C
07E0  46 4E 5E 8A AB D7 22 A3  : C3
07E8  6D 17 FA A6 A6 7A BF 87  : 8A
07F0  04 F4 83 8A FC A1 7E 30  : 50
07F8  1F 9D A9 20 EA 0F F8 D3  : 49
--------------------------------------
SUM:  49 E8 C6 03 BA 20 75 8A  DE2A

0800  9E 34 12 E3 56 11 00 E3  : 11
0808  9B D7 CC 76 EF AF A6 40  : 38
0810  D9 BA 54 A6 E7 DD 82 1A  : ED
0818  9E 8B 6B E0 9B 4D 5E 88  : 42
0820  1E 2F C2 BB 06 66 C5 F5  : F0
0828  F1 63 4F 8B BF 32 0D 74  : A0
0830  79 65 E7 33 ED 23 A5 2A  : D7
0838  8F 8A 9D 1A 98 EE 82 39  : 11
0840  0C 27 E3 69 AF AF 86 76  : D9
0848  FA BE 31 E7 42 93 23 42  : 0A
0850  1D 6B ED FA 3D 6B DB 87  : 79
0858  FE CF FE 9F EC C8 F0 D9  : E7
0860  ED 19 EC C8 F5 B3 7E 66  : 46
0868  BB 39 E8 86 E3 75 B8 8E  : 00
0870  71 37 18 96 F8 95 B2 B7  : 4C
0878  D8 95 C1 D6 A2 FF 13 71  : 29
--------------------------------------
SUM:  D9 0E DE 15 9D C4 EE C5  EB33

0880  EE EE 43 83 2C 22 3D 57  : 84
0888  C9 B8 58 67 D8 C4 39 7C  : 91
0890  02 4D A4 7A FB 72 67 48  : 89
0898  B1 26 73 21 95 57 38 38  : C7
08A0  8C D6 2A 69 A2 6E 45 C1  : 0B
08A8  D3 ED 54 70 D8 E7 F9 7C  : B8
08B0  CC 5E 49 77 81 F2 BB BF  : D7
08B8  1B BC BC 95 F6 81 47 F6  : DC
08C0  A5 7F 4E 1B 92 31 C4 43  : 57
08C8  11 D6 23 8C 4F A5 88 FE  : 10
08D0  57 DC 95 C9 05 B8 61 67  : 16
08D8  5F DB 85 80 C8 39 0E B1  : FF
08E0  0D 20 FD 73 6F 7B 48 DE  : AD
08E8  CD FF 8B 90 1D 89 67 43  : 37
08F0  17 EE F2 F1 7C 07 E1 A8  : F4
08F8  2E D8 49 C5 BB D5 37 6B  : 46
--------------------------------------
SUM:  3B E7 83 13 F6 1E D7 D2  55E6

0900  D2 1B 78 B3 76 7D FD 9D  : A5
0908  5B 51 37 56 D9 77 82 57  : 62
0910  D5 52 CC E4 B6 D3 1E 90  : 0E
0918  79 67 B6 4A 81 43 5F 04  : 07
0920  08 E1 E5 A8 C5 84 A3 4D  : AF
0928  33 50 FA 22 F6 C1 61 BA  : 71
0930  2D B5 C6 E5 63 45 3B 30  : A0
0938  E4 60 CC 38 F7 9C FA FB  : D0
0940  4D 0B 47 13 4B A3 88 78  : A0
0948  28 F5 BE E1 1A CD 54 8C  : 83
0950  00 FB 0D 57 16 BE 5B 55  : E3
0958  C2 DD E9 18 71 F8 32 E5  : 20
0960  88 D3 72 F7 51 1B 42 5C  : CE
0968  1F 3B E2 83 69 24 B8 CB  : CF
0970  47 94 49 0D 90 EE F2 48  : E9
0978  6E EA B1 C9 01 FF 8B 26  : 83
--------------------------------------
SUM:  5A CF EB D1 D2 82 15 8D  3AEF

0980  33 17 5C CC 9E C9 C7 33  : D3
0988  DC 23 2F 3E 0C 9F 16 4D  : 7A
0990  DF 3F 40 EB 26 D3 30 A6  : 18
0998  89 B5 25 3E 41 7A FB CC  : 23
09A0  C1 FA 3B C1 C2 39 1C FF  : CD
09A8  8B BD 4D DF B8 47 97 79  : 83
09B0  1C F0 FB 90 3C 48 C3 BD  : 9B
09B8  1F 64 54 1B 75 3B A3 F3  : 38
09C0  32 32 5B 78 59 1D 0A 4B  : 02
09C8  BC 8F 6C DC CD E8 05 D0  : 1D
09D0  9C 3A 27 D1 6C DF 6E 27  : AE
09D8  87 BE FE 6D F6 E1 90 61  : 78
09E0  B3 A6 88 B1 35 DD 26 41  : 0B
09E8  2A C6 64 51 AB 63 34 F2  : D9
09F0  4B D2 5C 88 1A 92 81 A4  : D2
09F8  39 51 15 14 86 A8 B7 E7  : 7F
--------------------------------------
SUM:  70 81 10 AE 44 F7 C0 7B  2F16

0A00  82 F9 8C 6A 0B 7D 69 A7  : 09
0A08  2F 55 A9 1D C4 EA C6 95  : 53
0A10  F3 36 1B 6C DE 3F 9B 8F  : F7
0A18  8B 7A B7 7F 2A D7 B5 C7  : B8
0A20  93 52 59 47 3D 59 18 12  : 45
0A28  F4 3C 1F 37 A1 E0 F1 BD  : B5
0A30  22 46 DD 4D 4E 76 F6 70  : BC
0A38  4B 75 18 05 E4 D2 72 4C  : 51
0A40  6A 46 74 FD 58 4E 75 03  : 3F
0A48  D9 74 E4 6A E1 14 3A 2D  : F7
0A50  8D BC 09 B6 C5 85 27 7F  : F8
```

```
0A58  8A 91 CE 4C 26 7A 09 AB  : 89
0A60  60 54 0A F8 CB 4A 62 BE  : EB
0A68  A0 B8 77 DE E8 26 65 C8  : E8
0A70  EB 57 AB AA FC 58 D3 CD  : 8B
0A78  B7 44 F1 43 B3 CA 4B F5  : EC
--------------------------------------
SUM:  1F F5 C0 6E 6D F1 B4 BF  029C

0A80  1F 74 BD DC C7 9C EA FF  : 78
0A88  8F B1 AC 07 FF 6C D8 75  : AB
0A90  77 8C 54 08 72 35 F0 00  : F6
0A98  E6 9D 2A B2 F0 BB A8 FF  : B1
0AA0  25 5E 78 39 EF EE 89 7A  : 14
0AA8  BB F1 97 A7 FA 30 40 F5  : 49
0AB0  FA 55 FF F1 37 A8 31 BC  : 0B
0AB8  89 9E 00 B5 43 E0 0B D3  : DD
0AC0  F0 55 D9 87 7E 77 1C 01  : B7
0AC8  EF CF EC 0F 72 57 9C 46  : 64
0AD0  AB 1E 7D D6 B3 79 7D 79  : 3E
0AD8  C7 B0 2D 49 B8 61 96 37  : D3
0AE0  43 86 1A EC B1 A1 21 B5  : F7
0AE8  63 05 6C 58 DF 97 BE E8  : 48
0AF0  26 66 85 32 F6 F0 E2 46  : 51
0AF8  63 8F C1 10 26 DF AF A4  : 1B
--------------------------------------
SUM:  EE 02 30 5E 92 4D 9A EF  F84B

0B00  D2 A3 3A 61 7E E8 5F 02  : D7
0B08  CD F9 7A 4C CD 46 E2 9A  : 1B
0B10  56 5A E7 29 03 49 56 3A  : 9C
0B18  42 4D 9B 81 38 59 59 97  : 2C
0B20  C0 C5 9C 98 50 A0 3F 7B  : 63
0B28  0A 33 2E 98 70 5B 52 2C  : 4C
0B30  2D 7A 72 4D A3 3F EF 81  : B8
0B38  73 A4 22 D8 1F D6 4C D3  : 25
0B40  98 4D 83 C4 6A D7 FB 52  : BA
0B48  76 23 92 47 6A 90 ED 6F  : C8
0B50  D0 79 CA D0 D0 53 5D 90  : F3
0B58  EB D2 33 D3 01 2E CF AB  : 6C
0B60  77 2F 26 87 4C 07 3F 43  : 28
0B68  AB EB 65 E4 07 DE 83 D7  : 1E
0B70  71 9A 36 59 FB 3A B2 B8  : 39
0B78  FC 21 A5 FE AD 62 F6 C4  : 89
--------------------------------------
SUM:  F9 E9 0C 1C A8 49 3A FA  EE5E

0B80  A6 8E 44 D5 DE AA 6C EF  : 30
0B88  2E 1A 63 51 9E 77 C4 5B  : 30
0B90  6F AB A7 97 91 30 C6 0B  : EA
0B98  97 B7 72 F2 23 52 7B E9  : 8B
0BA0  92 BA FA AF C3 C2 CA 2B  : 6F
0BA8  96 25 75 00 56 6E 34 20  : 48
0BB0  9F EE 3E 1C 1F 30 D5 7E  : 89
0BB8  9C 94 D8 AA 93 37 87 2D  : 30
0BC0  5C 6A 82 CF 8F 77 DA CD  : C4
0BC8  63 7D 04 CE ED 55 7D C9  : 3A
0BD0  29 AF 46 75 AE 74 42 F4  : EB
0BD8  BE 6A 2C 8D 66 7A A1 13  : 75
0BE0  94 B3 B9 B3 7F 3D A6 22  : 37
0BE8  D7 98 93 BA 62 66 9E 4A  : 6C
0BF0  AA 76 6F 0C DD 58 AF 73  : F2
0BF8  81 22 BF 44 1E 6F D1 C7  : CB
--------------------------------------
SUM:  79 4E B7 80 67 5E C9 77  193C

0C00  93 A5 9B F4 64 84 BF B9  : 27
0C08  4A 86 FC 86 39 E9 BB 7A  : A9
0C10  92 1F 5F 8F C3 B3 7E 38  : CB
0C18  CE DC 89 54 3B 72 8D 3A  : FB
0C20  82 E2 55 0C E6 75 DE 77  : 75
0C28  90 7A EF 7F B5 48 17 D4  : 60
0C30  86 31 22 4A DD BA 12 16  : E2
0C38  8B B3 24 44 52 2D 2B 7F  : CF
0C40  16 30 A4 F7 44 4F BB F3  : 22
0C48  DD 7E 30 C1 39 8D A5 25  : DC
0C50  98 0A 91 5E F5 DA 45 D1  : 76
0C58  57 AB F5 6C 9D BD BA 0C  : 83
0C60  EE B2 10 53 D3 BB 22 CE  : 81
0C68  9E EC 35 90 9B E9 DD 11  : C1
0C70  6F 4E E4 99 AA 9B B9 7D  : B5
0C78  00 8D 5B B9 79 26 8D 36  : 03
--------------------------------------
SUM:  3D 42 E7 2D 05 0E 5B 0C  93EB

0C80  72 6A 40 9C A0 72 03 5C  : 29
0C88  30 90 BC A8 1F 9D 0F 06  : F5
0C90  EF 3E 5F A0 3A 8C 9A 28  : B4
0C98  4D AA BF 4D E0 54 92 CD  : 96
0CA0  FA 6D 8D 0A 2D 9C BA E6  : 67
0CA8  86 CE 39 BC CC 66 C8 61  : A4
0CB0  D6 63 8F 2E 39 62 1B 73  : 1F
0CB8  8C 4D A8 81 ED FA 66 E1  : 30
0CC0  E3 E2 7E 89 DD 33 92 2D  : 9B
0CC8  00 F9 A1 88 9F 75 47 92  : 0F
0CD0  65 1B 21 51 E1 FB DA 11  : B9
0CD8  99 3A D9 CB 8E 6B 12 4C  : CE
0CE0  9F 13 31 D4 94 DA D7 A6  : A2
0CE8  88 09 F2 A5 C5 46 2B 70  : CE
0CF0  53 95 46 33 6A 0A 55 EF  : 19
0CF8  40 1A 5D 99 A0 66 FD 56  : A9
```

```
--------------------------------------
SUM:  5B C8 F6 18 46 EB 5A 69  2663

0D00  36 20 6A EA 48 4E 66 65  : 0B
0D08  1A D1 CA 2B BF CF 28 A5  : 3B
0D10  33 28 A3 AE F7 D3 BF F8  : 2D
0D18  C9 6A 68 6E 13 F2 67 22  : 97
0D20  1D 01 DB A4 C0 CB 19 0A  : 4B
0D28  B1 81 33 09 BB CC 6E 89  : EC
0D30  0E 4F EE 21 0E 21 EB 18  : 9E
0D38  D2 8E 77 DF 9B 5A D5 72  : F2
0D40  09 99 7E E5 5C 85 59 97  : D6
0D48  E3 2B 91 2D ED B9 15 A6  : 2D
0D50  AD DB 80 0E B4 76 BD 1D  : 1A
0D58  56 5D 8C 44 80 31 B7 94  : 7F
0D60  95 9B 50 D5 7A 89 5C AB  : 5F
0D68  EA 25 72 AF FE C4 AE 6B  : 0B
0D70  F6 A5 6A 1A 91 6B 66 A4  : 25
0D78  5C F6 31 18 26 B4 35 A5  : 4F
--------------------------------------
SUM:  BA 39 2A F8 E1 45 82 8E  C9B5

0D80  86 5F 27 3B 81 D3 7D E7  : FF
0D88  09 4E C4 3C 03 71 E7 A9  : 5B
0D90  02 BA 88 C1 C7 25 5F ED  : 3D
0D98  D0 95 8B 6D B1 BC A9 2C  : 9F
0DA0  D4 F2 A4 AE 46 7C BE 54  : EC
0DA8  96 08 B2 04 9D 32 23 C0  : 06
0DB0  26 9D 11 F0 8C BF 37 95  : DB
0DB8  36 A6 B8 24 7F 86 D5 5E  : F0
0DC0  62 95 AC D6 67 99 A5 A1  : BF
0DC8  39 58 6C 5B 33 CB 69 6F  : 2E
0DD0  02 EE 5B 03 D5 3D 8E CD  : BB
0DD8  58 79 BF 55 23 AA BF 89  : FA
0DE0  F9 0B B4 D9 BC D3 FA 7E  : 98
0DE8  09 27 3F FF F3 4B EE E3  : 7D
0DF0  90 7A BA D1 83 D5 3D A9  : D3
0DF8  0E 68 AC 32 BC AF 5B 0F  : 29
--------------------------------------
SUM:  BC A1 A8 CF 6A 05 34 2F  F862

0E00  5C E2 F4 95 2A 9E 39 72  : 3A
0E08  E7 86 5E 59 72 A5 51 93  : 1F
0E10  CE 95 E7 49 F5 A5 6B 9C  : 34
0E18  D4 2B 33 17 A2 DB D9 BF  : 5E
0E20  1D 7B 4D BD 3D 17 76 D6  : 42
0E28  9C 25 AC D6 FC 00 E7 9E  : C4
0E30  6E 1F A1 E5 2E 23 10 4C  : C0
0E38  CE CF 35 27 4F F0 A7 7C  : 5B
0E40  92 A7 71 65 19 A7 88 9A  : F1
0E48  B5 5E F7 9B BC 93 86 F0  : 6A
0E50  31 C0 7E F3 11 B3 A8 FC  : CA
0E58  7C 7C 5C 1E AB 7F B4 B5  : 05
0E60  21 38 78 F2 75 45 E1 AF  : 0D
0E68  01 59 3C 12 60 93 C5 0F  : 6F
0E70  FB A2 D3 FB FF EC C3 EC  : 05
0E78  7F 94 5E F7 F8 1F F7 61  : D7
--------------------------------------
SUM:  6A BE 62 F4 46 3C AC E2  04A8

0E80  BE E8 43 1E A5 8A 2E CD  : 31
0E88  67 55 BF 47 54 A0 3E 42  : 36
0E90  2D 94 B3 80 8D 21 AC FC  : 4A
0E98  CF 5D 1A 05 2C B3 0D 87  : BE
0EA0  4C 62 F0 36 1D 30 32 FA  : 4D
0EA8  6F 74 78 50 1F 87 67 B0  : 68
0EB0  E0 0A FD CB D8 2A 7A BC  : EA
0EB8  37 A1 E1 BD C0 F7 FD 60  : 8A
0EC0  AE BF A8 8D E9 EB 9C 7A  : 8C
0EC8  E4 63 B8 F2 5A E6 DE 40  : 4F
0ED0  96 F4 75 15 D1 F7 23 A5  : A4
0ED8  BD 64 40 E8 DB D8 B2 F3  : A1
0EE0  FA FF 26 5E 43 9C F3 9F  : EE
0EE8  F5 0A 7E B3 EE 16 98 9D  : 69
0EF0  65 61 15 CB EB BD E0 6E  : 9C
0EF8  53 97 D7 3D 93 AF D2 F7  : 09
--------------------------------------
SUM:  7F 2A BA 8D 24 94 C1 4B  6F98

0F00  DF D5 BF C2 54 BA 2C EE  : 5D
0F08  60 B3 53 FC D4 26 AE 5E  : 68
0F10  6C B5 C0 97 9B 96 4A 6A  : 5D
0F18  ED FE 17 A7 92 92 D6 A4  : 47
0F20  86 49 A3 B4 8F 49 3B A4  : DD
0F28  C2 22 A1 C3 6A 56 B1 CF  : 88
0F30  F3 90 E0 FA BF AC AC 0E  : 82
0F38  6C A0 6F 23 97 DF 76 5A  : E4
0F40  ED A8 96 9F A6 21 37 E0  : A8
0F48  FC D8 92 22 D3 EB 06 B3  : FF
0F50  A4 81 BB 72 10 7B 11 9F  : 8D
0F58  AD 48 8A BC 6A 2F F6 10  : DA
0F60  9B 1B 41 CA B7 55 6D AB  : E5
0F68  16 4E 39 66 39 3C 0E 56  : DC
0F70  09 C9 EA 1C 9E 41 E2 02  : 9B
0F78  4C EA 87 EB 2F 49 7E 41  : DF
--------------------------------------
SUM:  7F 3B D4 B6 54 03 27 BB  11C0

0F80  F0 05 EC 6D 0F 39 36 87  : 53
0F88  B3 B4 92 22 68 C1 F4 41  : 79
```

▶いま，ポルシェ（赤）が目の前でスクラップになっている。ばかやろー，こっちはカリーナでがまんしているのに高い車を捨てるんじゃない。でも，隣のOA機器捨て場でMZ-80Kを拾ったからいいや。
筒井　圭一朗(19)埼玉県

```
0F90  92 F5 9A 67 0D 29 74 11  : 43
0F98  F6 A0 94 3D 21 05 1F 8F  : 3B
0FA0  2B 1F 11 A9 37 9A 2B 3D  : 3D
0FA8  AA 64 55 DB 8F 69 B3 ED  : D6
0FB0  02 FB 7B 3D A8 A2 4F B4  : 02
0FB8  68 C4 7C 11 7B 1B 5E 34  : E1
0FC0  A2 D7 D9 F1 B1 0B 5B DE  : 38
0FC8  37 F3 90 BE 1A 3D 8B E2  : 3C
0FD0  D3 04 AB 77 A3 E6 09 D8  : 63
0FD8  EB 43 D5 FA C5 CE E8 FF  : 77
0FE0  28 3E F0 5E 56 B8 80 6C  : AE
0FE8  26 D0 69 EC E8 C6 9B 5B  : EF
0FF0  3C 5A 3F 29 97 9E 9A D6  : A3
0FF8  5E 7A 59 08 FB 6E 77 A3  : BC
------------------------------------
SUM:  E9 83 E3 A0 91 6E 4B 51  613B

1000  CC FF 62 F1 C8 67 F9 1C  : 62
1008  91 CE BB 14 85 E5 9E 73  : A9
1010  90 F1 50 CB 77 67 B5 63  : 92
1018  C2 63 8C DB F2 97 92 FC  : A3
1020  DA 5E C1 98 BF BC 42 1E  : 6C
1028  C1 BC D2 67 03 15 54 43  : 65
1030  91 AA 0A C3 C0 06 F3 95  : 56
1038  8C DD A9 C3 E0 27 58 59  : 8D
1040  93 D1 EE EC F6 A1 E1 D9  : 8F
1048  21 3E A9 E8 07 68 F2 07  : 58
1050  D3 B4 58 42 34 7A C6 AB  : 40
1058  9A 93 D8 B9 F0 08 B0 02  : 68
1060  AF CF 38 24 E8 42 FC FA  : FA
1068  8B 1E F4 4F 69 30 23 F6  : 9E
1070  CF 9F F0 F2 31 38 FF 9B  : 53
1078  FD 18 F8 65 F3 13 D4 8A  : D6
------------------------------------
SUM:  8E BC 1A C9 AE 90 FA DF  A319

1080  6C 85 D0 27 A8 A4 B3 DA  : C1
1088  C1 2B BF 68 0F 7F 8D C1  : EF
1090  14 AB 55 45 6A F4 E0 E6  : 7D
1098  CE 3D B9 F0 1F F3 59 90  : AF
10A0  CD 6C 67 7F 63 6A D7 4C  : 0F
10A8  4A 00 FA CC BE DC D2 FA  : 76
10B0  AE C2 13 CB 10 27 D3 75  : CD
10B8  95 8A 95 6F 79 DA ED AC  : 0F
10C0  7E D9 1E 5B ED 8B 40 D2  : 5A
10C8  6F DE 84 9A 60 B3 7B 69  : 62
10D0  4D 64 FF F3 AC C0 04 19  : 2C
10D8  DB CA BB 3D A8 C3 9C 22  : C6
10E0  E0 38 C6 8C F9 A8 9A A5  : 4A
10E8  FA 91 88 FB D7 89 B6 3B  : 5F
10F0  F5 16 CD 72 54 9D 14 4E  : 9D
10F8  BB 5D B5 46 79 C9 4C C7  : 68
------------------------------------
SUM:  08 71 D2 AD 28 A9 ED E3  EB71

1100  09 9F 68 54 3F 8E 58 05  : 8E
1108  55 DF 41 F7 96 67 95 66  : 64
1110  7C 00 F1 34 AF 1C 23 19  : A8
1118  76 89 C1 17 E8 83 82 F8  : BC
1120  F8 40 7E 1C 47 30 BD A5  : AB
1128  B9 17 9F 9B 49 63 AE 84  : E8
1130  BA B6 B4 A4 62 1A 9D FD  : DE
1138  37 9C A2 72 2B 7F 39 0F  : D9
1140  75 F9 D5 9E D3 CB 73 F4  : E6
1148  C5 E9 C8 E5 CE 84 07 B9  : 6D
1150  92 7D D1 27 59 33 07 4F  : E9
1158  E1 05 36 8B D5 13 36 9F  : 64
1160  38 29 3D 3E 6C 19 22 57  : DA
1168  C0 AC 27 50 9C E8 52 D1  : 8A
1170  2A 00 21 AB 1B 29 D4 4E  : 5C
1178  F0 91 F6 82 ED 31 46 D1  : 2E
------------------------------------
SUM:  B1 7A ED 53 68 B0 18 93  AF94

1180  8D D9 DE 13 26 29 CF 94  : 09
1188  6C E9 42 B0 83 8A 54 6A  : 12
1190  D1 46 A5 C8 73 3E F3 94  : BC
1198  DF 79 47 71 2F 21 90 92  : 82
11A0  E9 93 5A 14 B5 4B 4A 38  : 6C
11A8  2C FA E8 6E D3 9F 2D 98  : B3
11B0  10 F5 49 33 87 AD D2 38  : BF
11B8  22 CA FF 4E 6D 5D 1C 38  : 57
11C0  52 51 5F 0F 58 17 C8 8F  : D7
11C8  30 40 BD 54 1E EE DD 6C  : D6
11D0  EF 09 C0 64 B8 32 41 F6  : 3D
11D8  B3 D2 3C 25 05 7F A7 9A  : AB
11E0  20 EB FE 43 C3 0A CE CE  : B5
11E8  CF A6 76 CE B0 87 F0 15  : F5
11F0  A7 CD 7A AA BF D3 08 91  : C3
11F8  BE B3 DA 87 8A 2F 0D CC  : 64
------------------------------------
SUM:  68 4A 76 2D B6 4F 6B 2F  2A37

1200  A9 D9 FE F2 B0 8D 87 D3  : 09
1208  6F 11 87 1B 0E 91 50 6C  : 7D
1210  0F 87 C6 C4 E3 4A 47 93  : 27
1218  E7 00 71 35 01 7B 04 96  : A3
1220  7E 6C 3F 5B 0C EC 30 D9  : 85
1228  AE F8 E5 47 2B CD 93 36  : 93
1230  43 82 39 4F 9F 0F B4 2F  : DE
1238  61 58 E5 6A 3F 2E 5E 1C  : EF
1240  7E EF D6 FA 35 FD C7 D9  : 0F
1248  95 BE FA 97 9F 4F EC 07  : C5
1250  85 77 88 1E F8 6F 46 57  : A6
1258  FA F0 EB 87 FE 75 71 2F  : 6F
1260  2F 61 DB 6C 35 44 B7 8B  : 92
1268  0E 3E AB 28 DD C0 C4 09  : 89
1270  FD 61 FD F7 61 AE DA 97  : D2
1278  4D F5 36 CB A1 EF 5F 5F  : 91
------------------------------------
SUM:  F7 B8 FA ED 95 AA 15 B2  A3B2

1280  4C 17 FC CC FE 41 6C 7A  : 50
1288  C3 EC B1 4A 76 84 0F F0  : A3
1290  1F 2F E4 1E BC C3 C6 0D  : A2
1298  4B 7D 98 95 88 11 FF DF  : 6C
12A0  72 C2 46 D9 3D FB 24 66  : 15
12A8  9A D6 91 88 CD A6 2B EB  : 12
12B0  EE F8 39 9F F7 C8 9B 77  : 8F
12B8  B0 DD D6 8F AC A7 23 40  : A8
12C0  AE 80 2B 9F 0F B3 C1 6D  : E8
12C8  88 7D 5E 3B C8 FD 6F D4  : A6
12D0  79 C5 E6 DE 11 43 F5 B4  : FF
12D8  E2 DD 7A C0 E1 8F F5 9D  : FB
12E0  D8 CC B3 5D 95 67 E9 65  : FE
12E8  59 FA 23 64 A6 3B 71 31  : 5D
12F0  78 1C B7 C2 EA F1 AC 64  : F8
12F8  EF 45 F5 4B 3E B4 E3 E1  : 2A
------------------------------------
SUM:  4C E2 7A 9E 91 72 50 CB  024A

1300  C9 1E 32 EE 42 40 6B FD  : F1
1308  95 41 E0 9D 69 01 96 31  : 84
1310  FD C1 72 2A C6 AC 46 53  : 65
1318  D4 43 E4 59 ED 72 82 E8  : 1D
1320  8B A4 4A B0 50 93 78 CC  : 50
1328  BA 25 E9 4F D0 E5 B6 51  : D3
1330  9F 93 15 F5 81 C8 34 0B  : C4
1338  FF 36 B8 0B F6 0B 03 23  : 1F
1340  DF 28 5A 16 43 41 70 5C  : C7
1348  8F 8D ED 15 B5 FE CC D0  : 6D
1350  68 E5 46 F0 38 33 AF 8D  : 2A
1358  4C D0 35 F5 85 25 A1 1E  : AF
1360  38 FE 8F 25 3A 08 AB D8  : AF
1368  FE 51 D8 FF AF F2 A5 F3  : 5F
1370  D1 61 F8 78 7F 8B 87 9E  : D1
1378  D1 4C BE 7A E0 E5 5A 18  : 8C
------------------------------------
SUM:  0C 5B 47 33 F2 AB EB 0C  6D91

1380  75 79 09 0E 2A E0 A7 E7  : 9D
1388  44 59 B6 08 71 53 19 E4  : 1C
1390  8E 33 DE 6F A6 61 90 71  : 16
1398  7B D8 02 07 7F 97 8A FB  : F7
13A0  EB 35 31 5B 19 8E C7 7B  : 95
13A8  34 0C 55 76 59 9E F6 DE  : D6
13B0  02 8C 0F 7B 9D 98 C3 77  : 87
13B8  AC 06 CA 8E F6 97 F7 BC  : 4A
13C0  1D EF 90 67 82 2E 67 7B  : 95
13C8  F9 26 39 9D EF DE CA 46  : D2
13D0  F1 46 C8 B8 73 CE 0D B8  : BD
13D8  F2 21 62 F0 B1 76 D8 BE  : 22
13E0  EC 2C EB B5 DD 17 8A 57  : 8D
13E8  92 0F 16 88 BD 2B 69 D6  : 66
13F0  1B CB 86 67 7F BF 0F 1B  : 3B
13F8  5C B9 69 9D 9E 23 57 84  : B7
------------------------------------
SUM:  7D EB E1 53 11 FA C0 C6  083E

1400  06 CA E6 4B 10 87 73 5B  : 66
1408  8F 2C DA 73 C7 58 F8 38  : 57
1410  FB E4 B0 51 92 DD 95 7B  : 5F
1418  59 DB 01 B8 11 90 A4 67  : 99
1420  27 CC 16 AE D9 C7 87 C2  : A0
1428  49 83 66 FE 18 7A D2 F0  : 84
1430  F8 4F 0B D0 75 EE 1E C1  : 64
1438  78 70 73 80 EE 4C 3A F7  : 46
1440  72 42 F8 3D 97 78 68 B7  : 17
1448  77 25 DE 1C 3D 50 81 A3  : 47
1450  23 C7 3E 23 88 87 49 E1  : 84
1458  07 9E 85 B8 DF 1A E4 F0  : AF
1460  9B 2B 98 04 11 2E 06 DA  : 81
1468  8A B9 42 AB 0F 3A 8C AB  : B0
1470  59 5C 0E 08 EB CF 17 83  : 1F
1478  52 4D 9B 84 FC F7 1D 02  : D0
------------------------------------
SUM:  AC 1C 87 32 10 5E 31 14  D46C

1480  3C 68 DE 8D C7 03 82 3C  : 97
1488  25 A2 30 A3 0E 4E 73 AD  : 16
1490  8F 27 AB F5 FD 5F 01 16  : C9
1498  23 77 90 F4 6F C0 72 20  : DF
14A0  64 F0 AA 43 AD 1F 13 84  : A4
14A8  F1 C7 0F 19 E3 87 2E 3C  : B4
14B0  EC 33 F3 7B 69 5A F5 78  : BD
14B8  FD CB EB E2 2A 56 AE 68  : 2B
14C0  3F 2C 53 F4 DF EC FC B7  : 30
14C8  FC 7F 66 BF EF F8 EF FF  : 75
14D0  7E 99 19 BF 5B 10 63 36  : F3
14D8  15 9F 5D 5B 05 F5 6D 7D  : 50
14E0  3D 98 6B 1F B1 7D B1 1E  : 5C
14E8  87 B1 88 4F 3A 3A FF 3A  : BC
14F0  48 36 F1 E1 9F AD 5F 4C  : 47
14F8  4C 90 F2 56 D2 34 A6 07  : D7
------------------------------------
SUM:  77 4F E5 44 EE 47 BC D3  FFDA

1500  42 89 EA B6 DB E2 2C 1E  : 72
1508  2F 3B F9 51 2D E8 DC 6F  : 14
1510  FB 88 BE 94 79 6D 1B D1  : A7
1518  79 62 56 8F 18 A4 B4 C7  : F7
1520  B8 5F 4A 26 A9 2F 1F 4E  : CC
1528  5A 5C D7 3B E0 9D D9 DD  : FB
1530  18 43 EE 89 E8 F0 EF DB  : 74
1538  77 20 7A E6 8D E2 38 DF  : 7D
1540  4F C9 1F 05 6E C6 09 40  : B9
1548  B6 46 CD 4B A0 A3 D8 57  : 86
1550  99 4B 15 DB 9C 33 E9 88  : 14
1558  B0 DC 7B CF 6F 31 0A 1A  : 9A
1560  12 B8 3C 1E 9E 8C 39 19  : A0
1568  88 36 87 26 B0 43 FF 46  : A3
1570  41 63 E5 69 B0 8D 6F 2C  : CA
1578  37 B5 5C 31 FE 10 DA 19  : 7A
------------------------------------
SUM:  E6 08 00 D2 AC B2 4B E7  E91A

1580  3F A8 A0 19 2B D3 EF 07  : 94
1588  4B 94 8B C9 FF A8 D0 9E  : 48
1590  DE 8C 38 9F 9E C9 A5 A3  : F0
1598  2F 3C 7F B9 CF C3 A7 2F  : 0B
15A0  24 3A 46 90 4B C1 15 A3  : F8
15A8  D6 29 B3 30 B8 73 B3 0D  : CD
15B0  D8 31 29 1B 47 39 E0 4E  : FB
15B8  CA 73 25 93 30 DE 3F 4C  : 8E
15C0  92 CB D5 45 B2 DA 2D 20  : 50
15C8  53 AA 88 7A B7 04 9C 6D  : C3
15D0  30 1B 22 83 5E 20 F1 B2  : 11
15D8  08 BB 1D C3 D6 78 D9 0C  : D6
15E0  18 BD 72 1D B7 64 C2 09  : 4A
15E8  EB B1 B4 20 6E 09 A6 1E  : AB
15F0  18 A0 E2 F2 73 A3 BB DE  : 3B
15F8  86 85 FF B3 EE 77 9A 58  : 14
------------------------------------
SUM:  F1 E9 CC 8F 34 4F 42 69  0044

1600  CA FB CA 68 CA F7 94 B1  : FD
1608  95 F7 BA 56 AA F7 4E 7E  : 09
1610  83 9D 9B 9D A5 23 F8 6C  : 84
1618  8E 58 4C FA 7E D3 FF 65  : E1
1620  5D C9 E1 DF F7 03 9E FA  : 78
1628  F9 8D 02 4B 4E E2 13 3E  : 54
1630  12 37 5E 1A 86 E8 D2 59  : 5A
1638  0C E4 56 43 22 D5 B9 62  : 9B
1640  DF 5F 00 9D 31 6D E8 51  : B2
1648  E5 85 BD 21 6D D0 A3 E9  : 11
1650  05 B3 CB 6E 79 1E 30 5B  : 13
1658  A2 2D BE 99 1E 2D FC B0  : 1D
1660  BA D8 B7 67 D1 64 16 C3  : BE
1668  CB D2 D3 41 54 A9 8D 02  : 3D
1670  64 9B EF D1 66 1D C8 C3  : CD
1678  30 85 3A 9B 55 F0 C9 37  : CF
------------------------------------
SUM:  68 E6 FB B5 99 28 00 F7  7102

1680  E2 6C 09 37 B6 AA 23 C6  : D7
1688  E0 53 4E 95 60 1A E8 0A  : 82
1690  53 84 53 96 2D 34 0F 1D  : 4D
1698  55 2A 78 E8 A0 ED D1 A0  : DD
16A0  29 EC F1 4F 58 81 ED F8  : 13
16A8  1E C1 02 12 3C 74 54 0D  : 04
16B0  C8 70 EF 21 2B D2 5C 08  : A9
16B8  7A 02 99 D4 85 F7 11 49  : BF
16C0  37 A5 0A ED 2D 44 3F 4F  : D2
16C8  8E 1A F9 A2 C4 BB 02 92  : 56
16D0  6E 74 2B C1 B5 24 DA F0  : 71
16D8  AE D4 2F 8E C3 B6 D1 49  : D2
16E0  35 24 2B 89 14 8A FB F0  : 96
16E8  AE 28 5F 56 15 DB C5 A3  : E3
16F0  0A 63 26 49 AD 78 57 78  : D0
16F8  99 15 F8 21 5D F0 57 EA  : 55
------------------------------------
SUM:  5A 57 A2 C7 C3 49 F3 F2  9E1D

1700  85 78 00 1E D1 12 2C E1  : 0B
1708  87 0E 52 1F A0 69 A0 5E  : 0D
1710  AC 89 87 1D 9D C8 56 92  : 26
1718  75 FE 2E 14 B9 69 2C AA  : AD
1720  0E 76 AF 1D 65 5B 5B 53  : BE
1728  34 F1 D4 32 2A 64 1F 6C  : 44
1730  5C 7F 59 15 2F 62 1C 66  : 5C
1738  C8 AC F0 75 B4 29 31 0A  : F1
1740  67 82 EB 2C F0 E3 12 2B  : 10
1748  5C 15 E9 02 B5 7F DA 8B  : F5
1750  EB F6 A7 BF 74 3B BB E6  : 97
1758  C3 A0 6A B0 E8 2A 5E 0F  : FC
1760  3A 8D 29 7C 71 BE 99 16  : 4A
1768  9E DC 15 2D 91 EF 3E 44  : BE
1770  0B 01 40 BC E5 72 09 FE  : 66
1778  48 58 58 7C 0C 29 79 17  : 39
------------------------------------
```

▶僕はアセンブラなどまったくわかりませんが，特集「ゲームマネージメント」を読んでとても感動した。自分でゲームを作ることは素晴らしいですね。　池永　智哉(20)福岡県

```
SUM:  2F 8E 8E C5 2D 05 73 C4   1FE4

1780  47 1F 34 F1 BF 01 BC 60   : 67
1788  E8 A6 4C D4 37 B0 D9 C3   : 31
1790  21 3B A4 49 2E AA 0E 18   : 47
1798  F3 E3 0B BC 48 6B 02 FD   : 4F
17A0  20 64 9E 99 2A 57 39 E1   : 56
17A8  CC 61 B3 1D 08 F8 4E 52   : 9D
17B0  E7 31 D1 0B CF 4F 9E 5B   : 0B
17B8  AB 44 8C 3C AE BD C0 C9   : AB
17C0  5C 17 3F B7 01 E9 F8 E2   : 2D
17C8  BF 14 DC A9 43 64 3C 42   : 7D
17D0  EA 8F 6E E8 12 AF 65 26   : 1B
17D8  B1 96 41 34 C0 B3 E3 B2   : C4
17E0  C6 0A 7B F0 8C 5A C2 45   : 28
17E8  04 AF 4E 9A C0 B3 5A CB   : 33
17F0  18 29 D2 42 C3 7B 87 AF   : C9
17F8  5A 16 74 CE F0 D9 63 4A   : 28
-------------------------------------
SUM:  B3 65 B6 DD 30 31 0C 94   3A04

1800  C8 24 4E 30 60 36 5C 27   : 83
1808  2E 1E 38 0B AB 8D 75 E3   : 1F
1810  85 02 86 0B CC 4E 86 D9   : 91
1818  EC 93 63 0B 0E 1C 76 58   : E5
1820  C1 C1 4F A1 0F 56 24 2E   : 29
1828  F0 C5 83 80 F6 C5 CC EB   : 2A
1830  1E B7 4D A0 5D A7 0A E7   : B7
1838  E1 38 D8 2F 11 30 9B A6   : A2
1840  C0 0C 79 CD 09 0E A5 96   : 64
1848  34 27 09 73 C4 E4 4D E6   : B2
1850  19 C6 9F 13 38 4D F6 0C   : 18
1858  83 9D 59 00 9D 79 94 61   : 84
1860  99 C3 89 EB 16 69 0E 35   : 92
1868  6C 33 FE D1 E3 DC FA 59   : 80
1870  14 D9 BD DE 2C 2D A2 24   : A7
1878  55 F5 7E 47 9A 16 DC 7E   : 19
-------------------------------------
SUM:  15 A6 A2 75 B9 5F 64 FA   CC3C

1880  72 4D 3D FF 4E 0A A0 23   : 16
1888  C1 3F 41 B0 5C 93 01 D2   : B3
1890  4F 4D ED A2 5D 26 F4 DF   : 81
1898  AC F6 F9 88 92 9C AB 72   : 6E
18A0  E9 44 1A BA A9 BF B6 EA   : 09
18A8  2A BD 50 40 98 1F EA 9A   : B2
18B0  E7 1F FB 32 FD 23 DF B4   : E6
18B8  D0 39 DA 16 3D 68 C1 85   : E4
18C0  B3 2C 53 AF 2C 53 CE CF   : FD
18C8  64 04 52 AF 0A 74 88 AD   : 1C
18D0  0A 41 AF C1 DA 3D C3 4E   : E3
18D8  9C 7B 5E E9 B8 87 E7 F8   : 7C
18E0  12 62 F4 2F 5A C1 57 B7   : C0
18E8  93 13 A1 C6 92 C7 86 7E   : 6A
18F0  7C 35 F3 62 78 7B D6 60   : 2F
18F8  B8 1E F5 A7 8F B2 E7 72   : 0C
-------------------------------------
SUM:  8E DC D2 21 CF 08 1A CC   5663

1900  C4 19 5A 3E BB 41 ED BA   : 18
1908  A9 16 01 5C D0 79 42 D6   : 7D
1910  85 6B 09 38 1D 18 58 B9   : 77
1918  DA 1D B3 08 27 A6 05 0B   : 8F
1920  68 35 AB DC 11 52 42 AD   : 76
1928  B5 45 70 5B 9C 49 90 5B   : 95
1930  C0 83 5A 57 EA B5 66 90   : 89
1938  87 CB 96 7B F2 E3 9D 03   : D8
1940  93 C3 83 5F 29 61 1C ED   : CB
1948  20 ED 38 8A 82 65 73 77   : A0
1950  6B 47 A3 AA 2C 02 9E A8   : 73
1958  4C 52 58 16 4F CF 2B 78   : CD
1960  E3 D4 F0 1E 38 A3 22 5C   : 1E
1968  76 93 60 DF 4F FD AD 7C   : BD
1970  3D 33 2E B6 CE 23 2F 84   : F8
1978  DC B8 2F 68 1C 99 F9 F0   : C9
-------------------------------------
SUM:  0C 1A 85 A7 EF 9E B0 BF   2805

1980  76 93 DE 7A 5D 08 5B 40   : 61
1988  8A 16 81 E5 63 21 FD 70   : F7
1990  B4 3A A9 B0 4E E8 52 88   : 57
1998  14 CF A6 F4 AC 3F F4 31   : 8D
19A0  62 58 FB 1F 21 99 E9 8E   : 05
19A8  8F BA 42 E9 77 A6 49 B4   : 8E
19B0  3D 73 D6 70 AB 51 A5 0F   : A6
19B8  26 3A A9 20 34 08 6D 53   : 25
19C0  5A 14 E0 25 3B 0D B8 8B   : FE
19C8  87 AA 29 0A D2 A2 79 BF   : 10
19D0  6D D4 45 DE 43 C8 74 80   : 63
19D8  A9 E8 79 0F 1E D6 A6 BE   : 71
19E0  1C FB D6 6F 5D 46 65 0A   : 6E
19E8  B5 2E 68 56 B9 C1 49 E1   : 45
19F0  AC 4B 20 76 29 F9 F9 2C   : D4
19F8  58 AB 92 5F 6A 0F 2A 51   : E8
-------------------------------------
SUM:  E8 0A 21 51 48 44 FE FD   8121

1A00  1F 6E 43 30 B8 AF F4 FB   : 56
1A08  6F B9 1D 57 A5 27 72 1C   : F6
1A10  59 5C EC E9 5E B1 A0 71   : AA
1A18  EC 4B 96 1A A0 AD 2E C3   : 25
1A20  3C E7 B4 74 CB F7 78 B5   : 3A
1A28  53 61 39 6D 9C E1 8E BC   : 21
1A30  31 67 C2 9D 4D AC EA B6   : 90
1A38  E4 9C E3 38 C7 71 EB 57   : 15
1A40  71 9B 86 C6 BE AC 89 63   : AE
1A48  C6 C5 B2 A7 38 6F AF BE   : F8
1A50  66 EF CB 24 DF E9 FC C4   : CC
1A58  26 EB 73 24 93 8E 5B 87   : AB
1A60  85 DB 78 1B A5 B1 21 26   : 90
1A68  BD 71 E3 44 1F 57 0F 81   : 5B
1A70  54 7A C8 9E DB 3A 4A 57   : EA
1A78  D6 55 F5 FF C8 4A 5A D8   : 63
-------------------------------------
SUM:  A6 6E 02 F1 A5 47 72 0B   CDBE

1A80  30 5F C1 FE 41 F0 6A E8   : D1
1A88  68 6B FD 96 D8 C4 A5 D2   : 79
1A90  EB 1C F6 F0 E9 0E 5A 10   : 4E
1A98  DA C0 29 D0 A7 ED 79 85   : 25
1AA0  F2 25 A2 B8 D5 A4 AF 0A   : A3
1AA8  59 EB 4D B6 B9 80 86 3B   : 41
1AB0  3C 71 DA 75 10 E8 53 F6   : 3D
1AB8  83 C5 1A B1 39 D8 73 A9   : 40
1AC0  F3 B7 FA 51 2D 47 91 0C   : 06
1AC8  8A FD B3 B8 A4 0E 91 25   : 5A
1AD0  70 2B 82 9C DA 6B 02 90   : 90
1AD8  E0 0F 82 27 BE A1 76 81   : EE
1AE0  38 08 C7 69 1F 38 B9 CE   : 4E
1AE8  94 29 D5 44 68 E7 68 39   : C6
1AF0  D7 D0 AD A8 9B F5 AC 3D   : 75
1AF8  07 1E 80 67 74 FC 42 8F   : 4D
-------------------------------------
SUM:  DE F9 3A 70 7F 04 86 48   82C7

1B00  38 A3 C1 3A C8 92 87 B8   : 6F
1B08  0D 61 2E 43 4D 7E 29 28   : FB
1B10  11 5C 16 ED C3 1D F1 49   : 8A
1B18  D3 81 6F 6E 61 E0 AD 71   : 90
1B20  14 57 3D 94 45 82 4A 1B   : 68
1B28  FB DB 07 8A 6E 08 57 78   : AC
1B30  3E 01 30 AE 20 5F DD 85   : FE
1B38  77 CA 57 6A 15 F9 81 58   : E9
1B40  6D 0A AB D7 0A C3 67 55   : 82
1B48  6B 02 B8 A1 7E A8 2B 48   : 5F
1B50  2B A7 0C 0B D9 07 7A 76   : B9
1B58  87 D7 8E E0 72 C6 18 FB   : 17
1B60  30 9A 5E 29 33 EA 01 1F   : 8E
1B68  35 8F CD 4B DB 6A 74 3C   : D1
1B70  24 58 06 C2 3E BA 6C 8F   : 37
1B78  EC 21 CF 26 74 3C 0E BF   : 7F
-------------------------------------
SUM:  EC 0A 3C CD B4 71 60 C1   02D3

1B80  EF CB E6 8F 00 55 5C 77   : 57
1B88  32 61 EB C7 9F 68 36 FC   : 7E
1B90  FA 6C 59 33 0C 83 F3 43   : B7
1B98  C0 BA 44 BD B1 7B 6A 5F   : 70
1BA0  F6 C6 F8 BF 38 75 01 0F   : 30
1BA8  34 A1 13 61 89 D3 3E 19   : FC
1BB0  32 77 C9 FB 61 E0 3B 22   : 0B
1BB8  AE E9 30 DB 77 27 BD 04   : 01
1BC0  DD 06 52 6F 04 13 6C 0A   : 31
1BC8  99 D0 24 B2 93 E6 C1 D6   : 4F
1BD0  29 6D 5C F6 30 A6 42 3B   : 3B
1BD8  55 24 0F 26 42 A0 E8 EA   : 62
1BE0  70 25 EA E1 56 FD BC 20   : 8F
1BE8  CC 8B 35 55 8A 83 C9 0E   : C5
1BF0  AA AD 3A 40 49 BC 51 43   : 6A
1BF8  4C 98 6C EF 93 AA 4D 80   : 49
-------------------------------------
SUM:  0B 75 18 DE BA 2F A0 59   F1E4

1C00  64 3E 1A 47 4F 63 02 9F   : 56
1C08  96 55 3B 27 3D 20 E4 18   : A6
1C10  7B 9F B6 5D 34 70 55 42   : 68
1C18  AD D8 0E D2 BC F4 D6 C3   : AE
1C20  EE C0 97 9C 1D 4D C2 46   : 53
1C28  F4 F8 0F FD 9C 9C 90 E2   : A2
1C30  4B B7 3D F8 71 C5 A1 86   : 94
1C38  49 69 17 07 B3 0A 43 88   : 58
1C40  4E 4C 1A 40 F0 83 E4 B4   : FF
1C48  79 F8 89 34 88 BA 9D 0B   : 18
1C50  FF 65 52 B5 0E FE 72 2B   : 14
1C58  4B 5F CF 51 06 49 6F 8A   : 12
1C60  85 9F 0B 27 95 0B 3B 92   : C3
1C68  C4 2B 0D 17 41 0A 75 54   : 27
1C70  9F 81 03 B4 CA 39 9D 29   : A0
1C78  C8 44 DE 4F 24 A7 3E 86   : C8
-------------------------------------
SUM:  59 79 D0 F0 A9 18 34 FB   7081

1C80  4C F2 3C 42 91 F6 D3 08   : 1E
1C88  F2 79 A5 23 E0 85 9E 19   : 4F
1C90  48 FC 5C 99 D4 77 85 23   : 2C
1C98  C5 CA 23 F3 0A 47 BA 0B   : BB
1CA0  3F 54 A4 7B 7C 96 41 37   : 3C
1CA8  38 51 C2 7C 70 B2 5A E6   : 29
1CB0  83 C1 1F B9 B3 DA D3 E8   : 64
1CB8  4D D7 FC B1 43 42 E6 48   : 84
1CC0  68 07 AB 7D 7D 16 B6 9F   : 7F
1CC8  49 A6 D4 D5 F5 55 5D 4D   : 8C
1CD0  4E A2 A3 4E D8 89 6B 22   : CF
1CD8  D6 C4 77 6C EA D5 CC 37   : 3F
1CE0  1D 10 C3 E7 E9 58 6F 27   : AE
1CE8  09 EE 4F 9F 4E 75 A3 70   : BB
1CF0  DF 5D 93 2C FA 7C D0 77   : B8
1CF8  7E C0 42 FF 56 97 30 79   : 15
-------------------------------------
SUM:  EA 9C 61 0F EC 46 60 68   51BD

1D00  A7 B1 CF A7 E9 16 EF 9E   : 5A
1D08  7E 6C 7B 3B AD CD 07 6B   : 8C
1D10  90 E6 93 A7 E7 39 63 E2   : 15
1D18  04 D4 AA EA 62 2D C9 A3   : 67
1D20  69 26 8F 0C 7B F3 93 F8   : 23
1D28  FE FF 3F C6 6A A3 00 7A   : 89
1D30  8A FE 8B 55 D1 FF 16 AB   : F9
1D38  FC 7F 7F F7 F0 79 69 FA   : BD
1D40  49 5E 1C AF 33 13 AF C3   : 2A
1D48  EC B1 24 CA F2 A5 70 C3   : 55
1D50  CB 21 C4 01 19 CC 47 18   : F5
1D58  8E B1 20 E2 6E 30 ED 89   : 55
1D60  D7 92 6F 6B 5D FC BE 3F   : 99
1D68  FF 30 1F 24 94 A3 E8 EA   : 7B
1D70  0F DE 69 12 27 3C 98 0B   : 6E
1D78  0F CD 1F 2A EC F9 5A F3   : 57
-------------------------------------
SUM:  28 C7 99 B8 35 DF 1F F3   0CA7

1D80  FF BC DD 60 7F C8 23 1C   : 7E
1D88  53 D5 4E 97 21 5F E2 7D   : EC
1D90  2C 47 F8 9D C0 64 6F BF   : 5A
1D98  86 DF E2 2D A2 02 D6 A0   : 8E
1DA0  5A CB 6F F7 83 32 7D 79   : 36
1DA8  5F 66 57 0A 57 86 6C F1   : 60
1DB0  F1 16 D6 01 6B C0 2D CF   : 05
1DB8  FF 4F 8F F9 78 9F EE C3   : 9E
1DC0  D6 6D FB 7D B5 9E D7 EF   : D4
1DC8  9E F6 FA E6 9F 4F A7 3F   : 48
1DD0  4F A9 D4 E9 F3 0B 60 01   : 14
1DD8  6E BB 14 F9 73 FB 9F 9D   : E0
1DE0  F4 FC C3 F0 F5 9B 6F 37   : D9
1DE8  6B FE FD BB 9D CF 28 A5   : 5A
1DF0  F5 42 5D 50 43 71 3C F0   : C4
1DF8  55 91 56 96 9B FB BF 2E   : 55
-------------------------------------
SUM:  87 E1 80 92 E9 6D 5D BA   7E48

1E00  A6 A7 FF FB F0 0C 7F FB   : BD
1E08  8D DF 89 CF EE 19 4F FB   : 15
1E10  D2 C6 24 F8 0D 4B 84 0F   : 9F
1E18  0F DF 18 C5 0D 54 FD 61   : 8A
1E20  96 8E CA 35 3F E1 35 F7   : 6F
1E28  8D 29 FE 23 4F EE 55 7C   : E5
1E30  B8 83 DD 47 3A F7 F2 D7   : 59
1E38  F7 B1 4F F0 69 DF BD 98   : 84
1E40  FE E7 84 58 15 5E 1F FF   : 52
1E48  E7 F9 81 33 C5 FE C0 21   : 38
1E50  4B B6 C6 18 E0 5C 17 4F   : 81
1E58  92 C2 73 3C 5F EC 27 16   : 8B
1E60  4E 76 DF 81 60 52 B8 2E   : BC
1E68  9F B5 CC 53 A2 4F EC 27   : 77
1E70  E9 27 52 4E 94 7F 9F D2   : 34
1E78  9E 76 31 92 7F F2 0F 38   : 8F
-------------------------------------
SUM:  1C 36 24 A9 57 1F F7 2C   8273

1E80  49 C4 67 3F B2 73 F4 CE   : 9A
1E88  75 33 9D 28 FF 6B D2 B5   : 5E
1E90  3B F0 00 5F E1 93 FF F2   : EF
1E98  A7 EB 31 BA CC 66 CA 7F   : F8
1EA0  1B 73 14 FF B8 3F C7 FA   : 59
1EA8  5F CB FB FE AE 7C 84 6E   : 3F
1EB0  DC F9 39 E0 72 B3 1C CB   : FA
1EB8  73 68 D3 38 DD F9 FB 5C   : 13
1EC0  F0 39 59 8E 65 BB F1 CC   : ED
1EC8  7F F6 53 FC BF A7 34 3B   : 99
1ED0  56 DF 81 18 CD DA 9B FA   : 0A
1ED8  31 9B 7E 0D 0E D7 FF 4A   : 85
1EE0  7F 97 FF FE D4 EB 67 27   : 60
1EE8  03 2B E8 76 B3 81 64 C0   : E4
1EF0  4F 95 BE EE BE B8 6D 17   : 8A
1EF8  F5 E5 7A E1 EC 1E F1 E5   : 15
-------------------------------------
SUM:  25 56 1A 87 43 93 D9 B1   FB05

1F00  7E 80 F6 0D 03 72 5A 10   : E0
1F08  32 B8 3E 9C 4A 90 FB 99   : 32
1F10  FD 69 1B 12 78 05 B2 0B   : CD
1F18  2B 4B 7C E0 B6 71 27 48   : 68
1F20  C4 2D B1 01 2B 72 81 89   : 4A
1F28  3B 12 50 14 1B 12 A3 56   : D7
1F30  E5 21 D8 95 4A B0 2C CF   : 68
1F38  87 DA F2 91 89 6B 00 BA   : 92
1F40  B8 9A A2 33 0B 2B CB FD   : 25
1F48  71 7F 8D 99 F0 4A 8A 85   : 5F
```

▶おおっ，11月号の特集は袋とじだあ。と，２秒ほど怪しげな期待を抱いたが，それは乱丁であった。　松永　哲也(18)新潟県

```
1F50   98 56 21 E8 40 AE 8C 3E  : AF
1F58   89 9B 32 89 5D 94 5A 2A  : 54
1F60   91 83 A6 85 5A AF 25 30  : 9D
1F68   7D 36 33 99 8D 5A 05 2C  : 97
1F70   03 EB B2 E2 D1 02 BF E2  : F6
1F78   A5 7A 86 6B 43 73 B5 A1  : 1C
-----------------------------------
SUM:   43 4E 29 7E 27 4C 57 2D  8AA8

1F80   99 57 07 D5 A0 AD 99 86  : 38
1F88   D6 97 17 AE 53 2A AC 2D  : 88
1F90   D3 94 50 AE E8 FD 62 54  : 00
1F98   1B 20 FF 97 89 4E 0C 82  : 36
1FA0   B5 46 62 B9 4F A7 7C 04  : 8C
1FA8   A2 0E A3 29 BE 88 16 0A  : E2
1FB0   31 55 EA 14 B0 29 29 AD  : 33
1FB8   9C A2 AD 79 69 32 54 E5  : 38
1FC0   12 E6 C9 6C 32 91 6B 72  : CD
1FC8   99 38 D9 60 A6 09 64 52  : 6F
1FD0   3A A2 D6 B8 AA 99 D2 87  : 06
1FD8   3F 2B 32 AE 51 8F AC 53  : 29
1FE0   2C 9D 95 9C 56 E5 7F 33  : E7
1FE8   39 5C A1 79 A1 3E D1 4C  : AB
1FF0   82 D8 3E F7 65 76 22 36  : C2
1FF8   81 47 EE C5 47 36 1E 21  : 37
-----------------------------------
SUM:   0D F0 15 3A 00 3D 9F 9D  5FB1

2000   E2 C1 D8 BA 1A 60 FF 3E  : EC
2008   01 68 40 94 5F DB 1C A5  : 38
2010   1E 97 A3 09 67 29 97 66  : EE
2018   59 9D 53 22 AE E8 B0 F9  : AA
2020   D2 C4 1B 05 2B 38 94 8A  : 37
2028   30 79 E2 C5 48 DB 6C A4  : 83
2030   3D 9C 09 B7 A0 16 80 AB  : 7A
2038   BA 00 C4 F3 C5 44 CF 04  : 4D
2040   35 D0 13 5A 08 11 90 1A  : 35
2048   00 25 9E 0C 43 A5 02 9D  : 56
2050   29 40 B3 0A 59 01 1A F0  : 8A
2058   B6 BC 07 8E 2D AA 2D 35  : 40
2060   E0 15 C8 57 3A 55 F8 AB  : 46
2068   E0 95 E1 B9 2E 23 EF 3A  : 89
2070   2F AC CF BD BA 98 BC 62  : D7
2078   76 7B 29 8A D5 84 40 88  : C5
-----------------------------------
SUM:   CC F8 E4 42 2E AE 6D CA  E1B2

2080   DA 92 6A F2 B3 5A E5 18  : D2
2088   6C F7 46 90 EF 36 1F 69  : E6
2090   C0 95 91 49 E9 81 0E C0  : 67
2098   AD 95 99 60 86 63 99 17  : D4
20A0   DF 48 1F 74 A4 EE 82 9D  : 6B
20A8   E0 3C DE CC 09 4F 0F FE  : 2B
20B0   30 0F 7A 4A 75 4F 08 B5  : 84
20B8   5E B4 07 EA 80 AD E9 5A  : 73
20C0   8A F5 27 66 80 3C 65 3A  : 67
20C8   C2 6C D5 0C 9D DC AA 62  : 94
20D0   03 69 66 A8 03 8E 49 C0  : 14
20D8   57 46 1F EF C9 56 AB 5A  : CF
20E0   AC 39 79 F5 3E 32 17 51  : 2B
20E8   6E EF A2 43 BD 0F 12 C8  : E8
20F0   5F AA AB 42 FA 3C 48 31  : A5
20F8   6D AF 43 98 91 7C 5C 28  : 88
-----------------------------------
SUM:   8C 8B E2 BA 22 A2 FD 2A  9C5A

2100   5D 43 B8 BB 0F 17 D1 09  : 13
2108   D5 E3 21 22 FA EE C2 30  : D5
2110   7A 18 0C 2F 88 84 2B 34  : 38
2118   2E 2E EE 14 2B 61 82 83  : EF
2120   F8 F1 6E EB 64 48 8D 20  : 9B
2128   86 31 E2 21 16 FE 2D DD  : D8
2130   E8 23 CA 43 9A F2 D0 8F  : 03
2138   22 35 B5 73 BA FA D2 DF  : E4
2140   CE 54 16 FD 84 2A F7 6F  : 49
2148   FB 07 C6 42 AF D7 A8 0E  : 46
2150   0F 44 1E 49 41 F5 D5 BD  : 82
2158   8E 58 3C EE B2 60 3B 28  : 85
2160   0A 01 7C 50 36 50 1D B8  : 32
2168   B1 32 7F 3B BB 28 17 0E  : A5
2170   AC 6B B5 B9 40 F9 05 07  : CA
2178   0B B1 85 63 5B 60 6C 4B  : 16
-----------------------------------
SUM:   3A 2C 0D FF 3C 43 F0 D5  FD37

2180   C7 43 B4 B8 BB 85 71 83  : AA
2188   15 4C 6B E9 21 71 6D 1A  : CE
2190   EF 59 73 0D 32 98 CD FA  : 59
2198   88 5E 45 BD B0 87 1D CD  : 09
21A0   F2 49 18 87 D4 43 02 F3  : E6
21A8   2C BF 92 86 CB A3 77 96  : 7E
21B0   84 6B 98 9D 82 89 01 3F  : 6F
21B8   63 76 84 88 B6 D1 AF FA  : 15
21C0   2A 1E 60 AC 60 5E C8 35  : 0F
21C8   DF 69 08 76 D7 B7 17 F1  : 5C
21D0   4A B1 28 1E 72 17 30 EF  : E9
21D8   2F 7B 0B 78 B7 66 66 F6  : A6
21E0   68 58 03 A5 81 22 2C 78  : AF
21E8   72 26 20 FA 08 5B DC 5E  : 4F
21F0   5B 46 CA BF 92 84 2D 95  : 02
21F8   E5 EC 5B A7 51 AF AE EF  : 70
-----------------------------------
SUM:   F4 92 80 5A 61 97 49 8B  8240

2200   4C 0F 56 AD 0E B6 1D D4  : 13
2208   5E C2 18 3A 21 43 AB D7  : 58
2210   95 D7 3A B7 86 65 CE AD  : C3
2218   69 6F D8 6C AC D4 2F 39  : 04
2220   0B AD 96 02 96 D7 28 2F  : 14
2228   B1 52 FA F0 17 B2 52 DB  : E3
2230   04 2E A3 5D 82 B1 47 CF  : 7B
2238   AB A7 31 58 83 B3 30 16  : 57
2240   D9 60 7B 13 2E FD 54 22  : 68
2248   48 87 DA 03 46 50 CE A1  : B1
2250   04 A3 46 88 6B 75 41 0C  : A2
2258   B1 CC F1 73 FC 04 23 DF  : E3
2260   5E DC 5D 43 4C CC F7 3F  : 28
2268   AE 42 E6 35 B4 3B D8 A5  : 77
2270   29 B2 EF D0 BC B6 8B 76  : 0D
2278   A6 73 B2 DF 21 73 15 23  : 76
-----------------------------------
SUM:   C4 84 54 E9 CB 15 AB AB  6303

2280   65 B5 9C D1 37 84 66 20  : C8
2288   22 FC 24 14 63 A5 78 78  : 4E
2290   48 5D C5 ED 21 18 E8 46  : BE
2298   F0 05 62 2F 40 72 19 5B  : AC
22A0   1D 00 C6 68 65 2B A3 26  : A4
22A8   EE 9A 21 74 F2 1D DC 43  : 4B
22B0   0D 3B BC 40 A7 90 AF 61  : 8B
22B8   C8 32 BF 37 40 82 60 D6  : E8
22C0   CD 18 E4 2D B7 F7 E0 8C  : 10
22C8   1B EC B9 00 C7 EF E2 D8  : 30
22D0   4C 64 62 0D 9F 25 0B 8B  : 79
22D8   CA D0 C8 F7 66 BE C7 C4  : 08
22E0   31 1E CA D8 5C 69 52 3C  : 44
22E8   E4 22 98 79 57 F4 80 D0  : B2
22F0   76 11 76 46 2D CE 72 50  : 00
22F8   BD 8C 10 A9 12 37 68 69  : 1C
-----------------------------------
SUM:   E5 2F F8 C5 AE 38 AD 51  9F28

2300   11 5C 17 EB 82 29 4B 76  : DB
2308   71 92 11 23 3B B8 87 73  : 24
2310   18 D1 63 39 E8 A1 79 0E  : 95
2318   FE 29 A2 04 67 55 BA 11  : 54
2320   2B 82 06 AB EE 2F 06 6A  : EB
2328   C0 0F 55 4A 84 4B 97 21  : F5
2330   05 31 2F 0D 37 7B A4 2C  : F4
2338   36 5A CB 8B 9B D5 32 0F  : 97
2340   49 A5 33 E1 2C 1D 18 0F  : 72
2348   80 60 36 01 8D 14 CA 32  : B4
2350   50 B6 B9 BC 2E 64 35 FB  : 3D
2358   11 70 39 77 FA 5F 35 08  : C7
2360   7B 08 66 2B 9E 09 E9 B8  : 5C
2368   53 3C CF 6E 50 4E FB 04  : 69
2370   D6 FA 6E DD 08 91 6D AE  : CF
2378   22 45 D6 A9 7E 22 97 EC  : 09
-----------------------------------
SUM:   AE B2 56 0C A5 9F AC 68  A03D

2380   94 BF E0 0C BD D9 1A DE  : CD
2388   48 DB D6 9A 05 77 28 48  : 7F
2390   17 36 56 53 4A 82 10 21  : F3
2398   40 2E 57 D4 B1 09 15 BD  : 25
23A0   8C 6B 78 08 5C 6B EE 36  : 62
23A8   11 4D 9C 75 3C 84 2D E2  : 3E
23B0   DA DC 16 DE A9 E2 1B 08  : 58
23B8   76 A6 63 7A AD FA 1B 0B  : C6
23C0   EE D0 65 58 6E FC C6 68  : 13
23C8   47 8D DA 15 D9 8D D9 28  : 2A
23D0   45 91 0E F1 44 0F CC F5  : E9
23D8   D0 8B 79 0D 46 7E D9 00  : 7E
23E0   8C 97 63 A9 B1 B0 1E 94  : 42
23E8   0A 64 5D C6 52 EB 64 01  : 33
23F0   C8 5E 66 01 A9 BC 18 39  : 43
23F8   B6 08 CA D8 28 06 AD 04  : 3F
-----------------------------------
SUM:   7E 12 A6 55 50 19 43 86  47F3

2400   2E 23 5A 96 DE C5 E2 08  : CE
2408   47 B5 86 02 87 2D CF F2  : F9
2410   6C ED AE A2 5B C5 86 A4  : F3
2418   3D F4 EB 50 53 58 41 BB  : 13
2420   43 DE A1 63 09 C5 AC 69  : 08
2428   0A 39 73 F5 A8 21 72 10  : F6
2430   53 6C 13 E8 AC 3C 07 BB  : 64
2438   1B 44 10 AD 7D AD 6C A0  : 52
2440   D0 BA F1 E3 BA F2 18 5D  : 7F
2448   AA 11 6E EF EE 24 46 BB  : 2B
2450   52 FF 66 52 BD 3A 97 5E  : F5
2458   81 5C 2B 17 0A 5D 7C 17  : 19
2460   5D D6 EB 14 BB 04 2E D6  : F5
2468   BE 76 A5 DD 6D 83 8A FA  : 2A
2470   F7 EE A0 16 EC 8E 4A 08  : 67
2478   25 74 3B 0B 14 5A 43 83  : 13
-----------------------------------
SUM:   5D 54 0B C4 84 FA BF 15  5CE1

2480   E3 5A DF 24 A6 01 38 2E  : 4D
2488   DD 46 53 25 E8 39 E8 45  : E9
2490   C0 53 48 D3 E9 81 90 5D  : 85
2498   0F A0 C8 A2 79 C8 18 51  : C3
24A0   B3 2C 70 CC B0 A2 DE 9B  : E6
24A8   34 14 B6 B8 1A 0D B6 C2  : 55
24B0   E8 D1 5E 7C 00 63 F1 C2  : A9
24B8   E1 AE CF A1 7B 83 79 7A  : F0
24C0   A3 21 3D 90 5D BB 99 33  : 75
24C8   67 D0 85 D8 99 42 DA 21  : 6A
24D0   12 1E CA E6 E1 3B 75 12  : 83
24D8   B4 88 06 8D 71 70 63 5D  : 70
24E0   A8 5B D9 82 DE B4 76 B3  : 19
24E8   71 7F 0E E2 E4 C0 F0 4B  : BF
24F0   82 EE E2 58 45 98 AF DC  : 12
24F8   80 17 51 AE AE AF AE EE  : 8F
-----------------------------------
SUM:   2A C8 41 A4 32 7B D4 45  F058

2500   0D A4 D1 42 F8 BE F0 58  : C2
2508   46 0B C7 00 BD 6C 5B DB  : 77
2510   18 96 A6 BF C9 0B F6 30  : 0D
2518   F6 11 60 43 91 0E E8 C7  : F8
2520   3A 25 C6 23 34 F0 40 8C  : 38
2528   6C C7 A2 60 60 B5 CE D4  : EC
2530   25 64 87 61 60 3B B0 9A  : 56
2538   EE AD 0E B6 C0 19 0F 5B  : A2
2540   22 1C 7B 73 01 78 00 60  : 05
2548   45 E8 EF 43 A2 82 00 5C  : DF
2550   40 1D 52 8D 7F 5C 17 F5  : 23
2558   91 AE EF 4A 44 50 29 28  : 5D
2560   07 AE 05 D1 80 C7 00 2E  : 00
2568   C3 5A 98 57 B2 2D 0C 07  : FE
2570   05 0B C8 F2 2E 2E EF 54  : 69
2578   7A 3A 2F 05 0B 0B 07 91  : 96
-----------------------------------
SUM:   9B 6F DA 8A 94 0F 38 72  6897

2580   2E 5D DC 5E 47 08 E4 30  : 28
2588   3F 18 30 DC C0 06 03 C3  : EF
2590   02 57 56 E0 4E FA 29 B4  : B4
2598   B3 D1 72 10 77 60 A7 01  : 85
25A0   D2 DF E6 21 06 C2 0D E5  : 72
25A8   B0 66 22 DF 69 3E 1C 22  : FC
25B0   EE AD A3 EC 8D 75 81 75  : 22
25B8   1F 37 09 91 BE C4 6F 81  : 62
25C0   01 AE EB 43 4F FB E8 F1  : 00
25C8   E6 6A BA D2 E4 17 F0 A2  : 69
25D0   DC 8F 0B 82 DF BF 42 B6  : 8E
25D8   C2 BE 15 ED C5 C9 9E 1B  : C9
25E0   4C 10 D7 75 D4 3B 9B 91  : E3
25E8   6C 56 0A 19 E2 81 1B B7  : 1A
25F0   57 37 16 DB 01 C0 46 07  : 8D
25F8   DD 19 B9 96 00 8C 07 76  : 4E
-----------------------------------
SUM:   22 E1 FD 2A 14 43 8B CE  CF2B

2600   29 99 9B E5 B4 F0 90 B0  : 26
2608   7D 98 A7 E0 80 DB 28 F1  : 10
2610   06 01 0B 7F 8A 6B F7 B9  : 36
2618   5B 05 B4 C7 0C EB 94 ED  : 53
2620   68 88 17 5C C5 8B 1C D7  : A6
2628   57 05 D7 40 DF 64 46 B9  : B5
2630   32 FE 8A 74 00 FB AD 1E  : F4
2638   8A 85 0B CB D9 17 33 0F  : 17
2640   DB 1A 45 E6 42 6D 13 74  : 56
2648   2F 2F 2D A1 DD A4 6B BB  : D3
2650   34 1C D7 58 C2 7C 68 79  : 9E
2658   A2 EE D0 91 0D 47 AB 45  : 35
2660   B7 07 AA 5D 6E 8B 77 C0  : F5
2668   40 34 52 8D 12 F2 E3 50  : 8A
2670   5B F4 73 A1 7E F9 40 05  : 1F
2678   03 A3 C2 18 4B BE BD 83  : C9
-----------------------------------
SUM:   B7 6C CE F9 7E 2A 6D 89  76AF

2680   5D 74 A4 CF E8 FB 94 10  : CB
2688   BC C0 D8 45 91 77 16 E6  : 9D
2690   D8 23 90 BE 53 20 0E 80  : 4A
2698   68 62 EB AE 42 15 75 A5  : D4
26A0   6C C0 2F 2B 5C 40 30 03  : 55
26A8   1C EE C3 4B EB 90 D3 E4  : 4A
26B0   D7 77 48 27 16 F6 63 C7  : F3
26B8   D1 EE 4B 75 4C FB 49 80  : 8F
26C0   33 8F 31 C5 F5 14 88 5E  : A7
26C8   48 CA DE 07 A8 A7 1D 19  : 7C
26D0   41 FA 28 36 9D 61 6F DC  : E2
26D8   19 8B E9 F7 82 87 0D 37  : D1
26E0   0D 77 8E 84 5B 6B F3 3A  : 89
26E8   1D 46 95 04 86 D8 00 00  : 5A
26F0   00 00 00 00 00 00 00 00  : 00
26F8   00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:   88 67 BF 13 54 4E F0 0D  D2D9
```

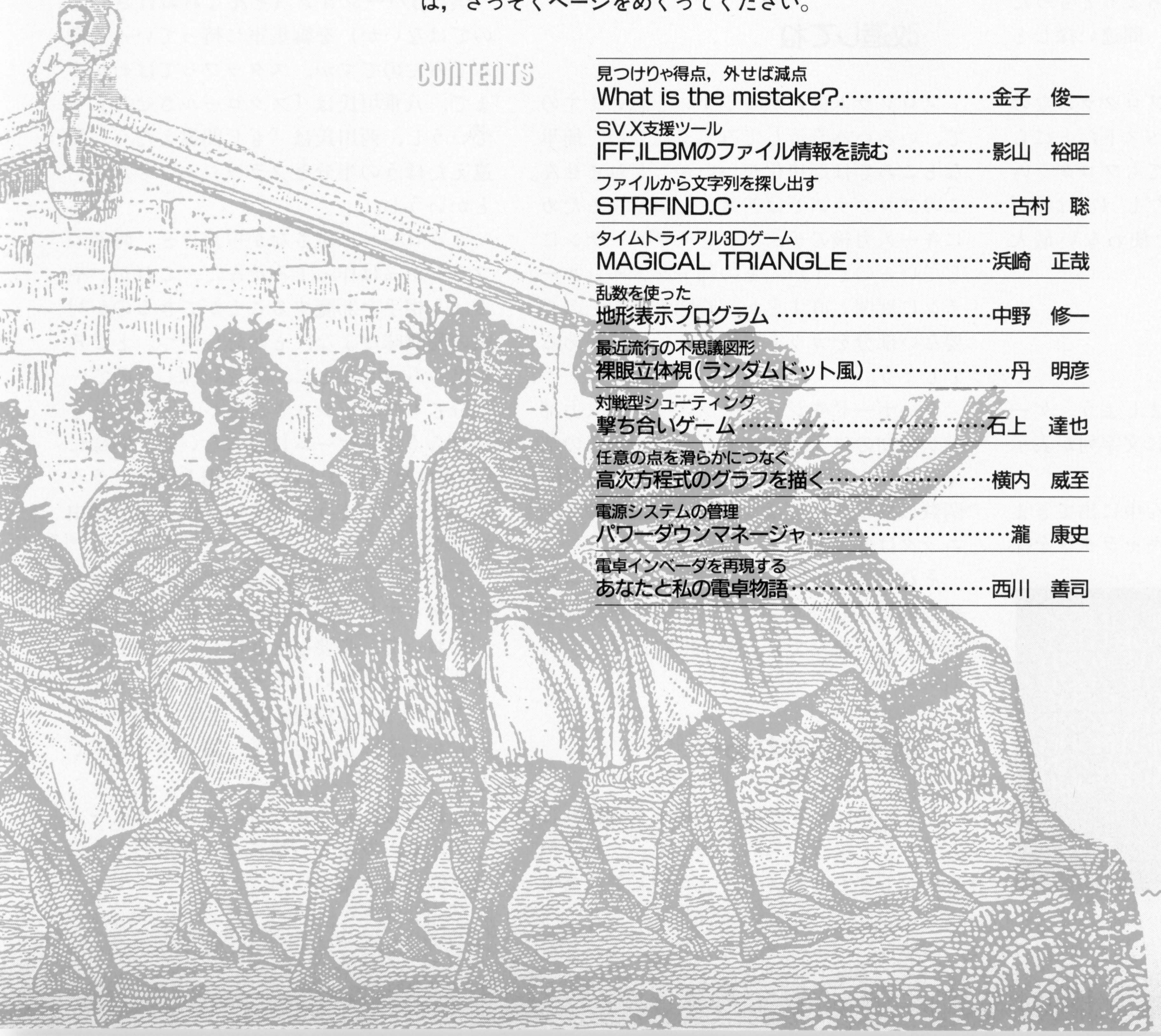
▶ 9月23日，２人目の娘が生まれた。バンザーイ。しかし，家はますます狭くなるばかり。男の僕とX68000もますます隅に追いやられそうです。わははは……う～ん。

松井　宏昌(30)兵庫県

Oh!X5周年記念特別企画

ショートプロ大集合

Oh!Xとなって５年間。基本的なスタイルを変えず，読者とともにすごしてきました。振り返ってみると，専門的なことが中心になっていて，気軽に楽しめるものが少ない，といえなくもありません。
そこで今回は，息抜きができるようなショートプログラムを集めてみました。ちょっとしたツールや懐かしい感じのするゲームなど，それぞれのライターが思いを込めた短いプログラムたちが10本。短いなりにもひとつの完成したプログラムであり，また，未完成ともいえるプログラムです。一見，貧弱に見えるところがあるかもしれませんが，そのなかにはいろいろなエッセンスが詰まっています。
日頃，市販されたプログラムばかりを使っている人も，気に入ったプログラムを見つけて打ち込んでみてください。
そして，ただ遊んだり使ったりするだけでなく，ユーザーの手の入り込む余地を見つけてプログラムをいじくりまわしてみましょう。きっとそこから，プログラミングという楽しみが見えてくるはずです。それでは，さっそくページをめくってください。



CONTENTS

見つけりゃ得点，外せば減点
What is the mistake?.……………………金子　俊一

SV.X支援ツール
IFF,ILBMのファイル情報を読む …………影山　裕昭

ファイルから文字列を探し出す
STRFIND.C……………………………………古村　聡

タイムトライアル3Dゲーム
MAGICAL TRIANGLE……………………浜崎　正哉

乱数を使った
地形表示プログラム …………………………中野　修一

最近流行の不思議議図形
裸眼立体視(ランダムドット風) ……………丹　明彦

対戦型シューティング
撃ち合いゲーム…………………………………石上　達也

任意の点を滑らかにつなぐ
高次方程式のグラフを描く……………………横内　威至

電源システムの管理
パワーダウンマネージャ………………………瀧　康史

電卓インベーダを再現する
あなたと私の電卓物語…………………………西川　善司

見つけりゃ得点，外せば減点

Where is the mistake？

ショートプロでは由緒正しい2人プレイゲームの登場
ひとりでもタイムトライアルゲームで遊べるけど，やっぱり対戦が楽しいね
ところで，よくある間違い探しで使える視線交差がこのゲームでも使えるのかな

Kaneko Shunichi 金子　俊一

さて，私がお送りするショートプログラムは「間違い探し」です。

X1のBASICだったら，きっとどれでも動くでしょう（リスト1）。CZ-8FB02の場合は，STICK関数が使えるようにNEWONを設定してください。解像度にも関係なく動くと思います。また，同じ内容のものをX-BASICへ移植したものがリスト2です。

本当は画面中に「WALLY，WOLLY，WORRY」などと表示して，ウォーリーを探せというプログラムを作ろうと考えていたのです。こりゃなかなかいいアイデアだぞ，などとひとりほくそえんでいたところ，コーディングする寸前に似たようなアイデアのプログラムがあることを知り，すっかりやる気がなくなってしまったのです。

とはいっても捨てるにはちょっともったいないアイデアだったので，間違い探しとして復活させました。

なんといってもショートプログラムなので，キャラグラでいくのがベストだとにらんだわけです。PCGを使ってもプログラムは大きくなってしまいそうだし（実は絵が描けないというのがPCGを使わない最大の原因だったりして）。

ルールは簡単

ゲームの説明をしておきましょう。ゲームを始めると画面の右と左に文字列が表示されます。

カーソルが文字列のど真ん中に出てきますので，左右で違っているキャラクタを探してカーソルを合わせてスペースキーを叩いてください。カーソルは無限ループになっているので，上端から上にいこうとすると下から現れます。左右も同様につながっています。得点は1回目が20点，2回目が40点，3回目が60点です。2人同時に見つけた場合は同じ得点が加算されます。

9×9の文字列の中に間違いが3つほど用意されています。3つの間違いを見つけるとゲームが終了します。と同時に次のゲームがあっという間に始まり，これがエンドレスに続くようになっています。

ひとりで遊ぶときは時間ゲームとして遊ぶしかないと思いますが，対戦も用意してあります。できれば友達と勝負したほうが面白いかもしれません。

改造してね

プログラムはわかりやすいと思いますので，いろいろ改造してみてください。簡単なところでは高速化がいいかもしれません。このプログラムではリストを短くするためにキー入力後のチェックをサブルーチンにしているのですが，その部分をメインルーチンに展開し直すとか，時間表示などの必要ない部分をカットするとかの方法があるでしょう。

キーボードやジョイスティックの入力部分が最初のほうで設定できますが，その部分も数値として直接STICK関数やSTRIG関数に放り込んだほうが速い，などのテクニックは基本的なところでしょう。

それから、X1のBASICでプログラムを組むのは久しぶりですので，多少無駄なコーディングをしているかもしれません。あしからず。

ほかにもやれるぞ

たとえば，一定時間だけ相手を金縛りにしてしまうようなものとか，カーソルの色を白にしてしまってカーソルがどこにあるのかわからなくするとか，なにかアイテムをつけてみるのはグッドだと思います。

スコアだけではなくって，一発逆転の要素があってもいいだろうし，ラッキーカードのようなものがあっても面白いかもしれません。ここいらへんまでくると，ちょっとBASICでは重くなってしまうとは思いますが。

ほかにも文字をすべてカードとして考えると，文字をすべてひっくり返しておいて，神経衰弱をしながらの間違い探しとか。

さらに発展させると潜水艦ゲームに近くなってしまうかな。

余談

実はβバージョン（そんなおおげさなものではないが）を編集室に持っていって遊んでみたのですが，スタッフってばわがままで，八重垣氏は「スクロールさせろ」っていうし，西川氏は「もし間違えたら，間違えたほうの半分をフラッシュさせたら」とかいうし。

スクロールは試さなかったけど，画面のフラッシュは作ってみたんだ，これが。やっぱり相手を邪魔することができるのが対戦の醍醐味だよな，とか思いつつ。ところが笑っちゃうことに，間違い探しなので，半分でもフラッシュされちゃうとどっちも見づらくってゲームにならないの。同様に「間違いを見つけられたら，相手側をフラッシュさせてしまう」っていうのも同じ理由でボツ。

グラフィック画面の画面半分ずつを違う色で塗っておき，ふだんはパレットで黒にしておいて，必要なときだけパレットを切り替える，っていうアイデアを思いついたときはあまりの素晴らしさに「膝ポンもの」かと思ったんだけど，情けない結末でした。

それでは，せっかくのエンドレスゲームですから，友達と一緒になって猿のように遊び倒してくださいね。

リスト1(X1/turbo用)

```
10 ' Where is the misstakes.
20 WIDTH 40 :CONSOLE 0,25 :CLS 4 :CLEAR :CLICK OFF
30 P1=0 :P2=1 ' 0=keyboad, 1=Stick1, 2=Stick2
40  DIM S$(8,8),D$(8,8)
50  COLOR 7 :LOCATE 12,23
60   PRINT "Just a moment."
70    SCREEN 0,1 :LOCATE 0,0 :CSIZE 3
80   A$=STRING$(20*9," ")
90  PRINT #0 A$
100 '
110 LOCATE 0,0  :CLS
120 FOR I=0 TO 8
130   FOR J=0 TO 8
140    S$(I,J)=CHR$(INT(RND(1)*26)+&H41) :D$(I,J)=S$(I,J)
150     LOCATE I*2,J*2
160    PRINT#0 S$(I,J);
170   NEXT
180  PRINT :PRINT
190 NEXT
200 '
210  FOR I=0 TO 2
220   X=INT(RND(1)*8) :Y=INT(RND(1)*8)
230    IF S$(X,Y)=D$(Y,X) THEN 220
240   D$(Y,X)=S$(X,Y)
250  NEXT
260 '
270 FOR I=0 TO 8
280   FOR J=0 TO 8
290     LOCATE 22+I*2,J*2
300    PRINT#0# D$(I,J);
310   NEXT
320 PRINT :PRINT
330 NEXT
340 ' INPUT
350 SCREEN 1,1
360 X1=4 :Y1=4 :X2=4 :Y2=4 :TIME=0
370  GOSUB 830
380 WHILE ATARI<3
390  COLOR 7:LOCATE 18,22 :PRINT TIME
400  OX1=X1 :OY1=Y1 :OX2=X2 :OY2=Y2
410   S=STICK(P1)
420     X=X1 :Y=Y1
430       GOSUB 660
440     X1=X :Y1=Y
450   S=STICK(P2)
460     X=X2 :Y=Y2
470       GOSUB 660
480     X2=X :Y2=Y
490  IF OX1=X1 AND OY1=Y1 THEN 520
500    COLOR7 :LOCATE OX1*2,OY1*2
510     PRINT#0 S$(OX1,OY1);
520    COLOR3 :LOCATE X1*2,Y1*2
530   PRINT#0 S$(X1,Y1);
540  IF OX2=X2 AND OY2=Y2 THEN 570
550    COLOR7 :LOCATE 22+OX2*2,OY2*2
560     PRINT#0# D$(OX2,OY2);
570    COLOR4 :LOCATE 22+X2*2,Y2*2
580   PRINT#0# D$(X2,Y2);
590  '
600  S1=STRIG(P1) :S2=STRIG(P2)
610   IF (S1=0) AND (S2=0) THEN WEND
620    GOSUB 720
630   WEND
640  ATARI=0 :PAUSE 30
650 GOTO 70
660 ' Key check
670  IF S=6 THEN X=X+1 ELSE IF S=4 THEN X=X-1
680  IF S=2 THEN Y=Y+1 ELSE IF S=8 THEN Y=Y-1
690  IF X>8 THEN X=0 ELSE IF X<0 THEN X=8
700  IF Y>8 THEN Y=0 ELSE IF Y<0 THEN Y=8
710  RETURN
720 ' Atari check
730  AT=0 :GOSUB 870
740  COLOR 7
750   IF AT=0 THEN 830
760    IF AT=1 THEN D$(X1,Y1)=S$(X1,Y1) :GOTO 810
770     IF AT=2 THEN S$(X2,Y2)=D$(X2,Y2) :GOTO 820
780    IF (X1=X2) AND (Y1=Y2) THEN S$(X1,Y1)="@" :D$(X1,Y1)="@"
:ATARI=ATARI-1 :SC2=SC2-20 :GOTO 810
790   D$(X1,Y1)=S$(X1,Y1)
800  S$(X2,Y2)=D$(X2,Y2)
810  LOCATE 22+X1*2,Y1*2 :PRINT #0# D$(X1,Y1)
820  LOCATE X2*2,Y2*2 :PRINT #0 S$(X2,Y2)
830   ' Score print
840   LOCATE 6,22:PRINT "SC=";SC1
850   LOCATE 28,22:PRINT "SC=";SC2
860  RETURN
870  IF S1=-1 THEN GOSUB 890 ELSE 960
880  IF S2=-1 THEN 960 ELSE RETURN
890   LOCATE 6,20
900    IF S$(X1,Y1) <> D$(X1,Y1) THEN 930
910     COLOR 6 :PRINT "HAZURE" :SC1=SC1-10
920      RETURN
930       COLOR 2 :PRINT "ATARI!" :ATARI=ATARI+1 :AT=1
940      SC1=SC1+ATARI*20
950    RETURN
960   LOCATE 28,20
970    IF S$(X2,Y2) <> D$(X2,Y2) THEN 1000
980     COLOR 6 :PRINT "HAZURE" :SC2=SC2-10
990      RETURN
1000      COLOR 2 :PRINT "ATARI!" :ATARI=ATARI+1 :AT=AT+2
1010     SC2=SC2+ATARI*20
1020    RETURN
```

リスト2(X68000用)

```
10 /* Where is the misstake. for X-BASIC
20 /*           Translate M.Hamazaki
30    int i,j,x,y,game_f,ata,all_time,ef=0
40    str befor_time,wk
50    dim int pl_x(1),pl_y(1),old_x(1),old_y(1)
60    dim str FIELD(1,8,8)[1]
70    dim move_xy(19)={
80       +0,0,-1,1,0,1,1,1,-1,0,0,0,1,0,-1,-1,0,-1,1,-1}
90    dim fst(3)={3,10,133,10},ccl(1)={13,11},sc(1)
100    screen 0,1,1,1:color[,1986,&H7FE]:sc={0,0}
110    while ef=0
120    GAME_INIT()
130       while ata<3
140         PUSH():MOVE_CUR():TIME_COUNT()
150       endwhile
160    endwhile
170    end
180 func PUSH()     /* 判定
190    int trig,t,x,y,chk=10
200    dim int cxy(3),han(1)
210    for i=0 to 1
220     t=i xor 1
230     x=pl_x(i):y=pl_y(i)
240     cxy(i*2)=x:cxy(i*2+1)=y
250     trig=strig(i+1):han(i)=0
260     if trig<>0 then { locate 11*i+5,12
270      if FIELD(i,x,y)<>FIELD(t,x,y) then {
280       if han(0)=1 then { sc(i)=sc(i)+ata*20:ata=ata+1
290       } else {ata=ata+1:sc(i)=sc(i)+ata*20}
300       fill(x*13+fst(t*2),y*20+fst(t*2+1)+3,x*13+fst(t*2)+11,y*2
0+fst(t*2+1)+21,0)
310       beep:color 1:print "ATARI ":han(i)=1:SCORE_PRT()
320      } else { beep:color 2:print"HAZURE"
330            sc(i)=sc(i)-10:SCORE_PRT() } }
340    next
350    if han(0)=1 and han(1)=1 then {
360     if cxy(0)=cxy(2) then {
370      if cxy(1)=cxy(3) then {  /*同一位置チェック
380       x=cxy(2):y=cxy(3):FIELD(0,x,y)="@":FIELD(1,x,y)="@"
390       chk=0:ata=ata-1 } } }
400    if chk<>0  then {
410      for i=0 to 1
420        t=i xor 1
430        x=cxy(i*2):y=cxy(i*2+1)
440        if han(i)<>0 then {
450        symbol(x*13+fst(t*2),y*20+fst(t*2+1),FIELD(i,x,y),2,2,0,
15,0)
460        FIELD(t,x,y)=FIELD(i,x,y) }
470      next }
480 endfunc
490 func MOVE_CUR()  /* 移動
500   int stw,x,y
510   for i=0 to 1
520    stw=stick(i+1)
530    old_x(i)=pl_x(i):old_y(i)=pl_y(i)
540       pl_x(i)=pl_x(i)+move_xy(stw*2)
550       pl_y(i)=pl_y(i)+move_xy(stw*2+1)
560       if pl_x(i) > 8 then pl_x(i)=0
570       if pl_x(i) < 0 then pl_x(i)=8
580       if pl_y(i) > 8 then pl_y(i)=0
590       if pl_y(i) < 0 then pl_y(i)=8
600       x=pl_x(i):y=pl_y(i):wk=FIELD(i,x,y)
610   symbol(x*13+fst(i*2),y*20+fst(i*2+1),wk,2,2,0,ccl(i),0)
620   if stw<>0 then {
630    x=old_x(i):y=old_y(i):wk=FIELD(i,x,y)
640    symbol(x*13+fst(i*2),y*20+fst(i*2+1),wk,2,2,0,15,0) }
650   next
660 endfunc
670 func SCORE_PRT()
680      color 3
690      locate 3,13:print "SCORE:";sc(0)
700      locate 19,13:print "SCORE:";sc(1)
710 endfunc
720 func TIME_COUNT()    /* 時間カウント
730      color 3
740         if befor_time <> time$ then {
750            all_time=all_time+1:befor_time=time$ }
760      locate 12,14:print using"####";all_time;
770      print "秒"
780 endfunc
790 func GAME_INIT()     /* 初期化
800     int chk
810     wipe():cls:home(0,0,0)
820     for i=0 to 8      /*文字列の初期化
830       for j=0 to 8
840         FIELD(0,i,j)=chr$(int(rnd()*26+&H41)):FIELD(1,i,j)=FIELD
(0,i,j)
850         symbol(i*13+3,j*20+10,FIELD(0,i,j),2,2,0,15,0)
860       next:next
870     for i=0 to 2      /* 間違い場所の設定
880       chk=0
890       while chk=0
900         x=int(rnd()*8):y=int(rnd()*8)
910         if FIELD(0,x,y)<>FIELD(1,y,x) then FIELD(1,y,x)=FIELD(0,
x,y):chk=1
920       endwhile
930     next
940     for i=0 to 8      /*プレイヤー2の文字列表示
950       for j=0 to 8
960         symbol(i*13+133,j*20+10,FIELD(1,i,j),2,2,0,15,0)
970       next:next
980     pl_x={4,4}:pl_y={4,4}
990     old_x={4,4}:old_y={4,4}
1000       for i=0 to 1
1010         x=pl_x(i):y=pl_y(i)
1020         symbol(x*13+fst(i*2),y*20+fst(i*2+1),FIELD(i,x,y),2,2,0,
ccl(i),0)
1030       next:SCORE_PRT()
1040     ata=0:befor_time=time$:all_time=0
1050 endfunc
```

SV.X支援ツール

IFF,ILBM形式ファイルの情報を読む

Kageyama Hiroaki 影山 裕昭

AMIGAで作成された画像データを再生する目的で作られたSV.X
そのSV.Xになかった，画像ファイル情報を公開するのがこのプロラムです
ユーザーに対するちょっとした心配りが，ユーティリティには必要ですね

創刊10周年記念PRO-68Kで，AMIGAで作成したアニメーションや静止画をX68000で表示するビュアを発表しました。現在，AMIGAでは，静止画，アニメーションともに面白いものがたくさんあり，これらの資産をX68000でも活用しよう，ということを目的としていました。

ここで，現在のX68000でのアニメーション作成環境を考えると，有名どころではDōGA CGAシステム，BBSではANIMAXというものもあるようです。あちこちのBBSに散らばっているものを揃えれば相当な数になると思いますが，本格的な使用に耐えうるものとしてはDōGA CGAだけという現状でしょう(断言)。それでもCPUは最高速で16MHzと，3Dレンダリングした物体を何十枚も作成するにはかなりのパワー不足であることは否めません。

それに対してAMIGAではソフトウェアメーカーが作成した3Dレンダリングツールが豊富なうえに，作成した画像をアニメーションする機能もあり，さらにCPUも68020，68030の高速なものが搭載されたものが用意されています。それに加えてアニメーションファイルのフォーマットがある程度規格化されています。

この差がAMIGAとX68000でのCGアニメーションの質，量の差となって表れているのでしょう。

プログラムの説明

創刊10周年PRO-68Kに収録したSV.Xは，ハードウェアスペック的にX68000でそのまま表示できないものを，自動的に画像を変換して表示するようになっています。私はプログラムを作った本人なので，変換が必要なデータの区別がつくのですが，一般の人にはまずわからないでしょう。変換されたデータがもともとはどういった形でAMIGAで表示されていたのか知りたいという人が，少なからずいるだろうと考えます。

今回発表するプログラムは，AMIGAで作成したIFF，ILBM形式のファイルに含まれている画面情報や，アニメーションの総フレーム数，SV.Xで表示するのに画像データの変換が必要か，といった情報を表示するものです。

プログラムはX-BASICで作成しました。少し長めですが，頑張ってリストを入力してみてください。450～790行は似たような命令が並んでいるので，うまくすれば入力の負担を軽減することができると思います。

RUNで実行すると画面には，

Input filename :

と表示されますので，IFF, ILBM形式で保存されたファイルのファイルネームを入力してください。ファイルが見つからないときは，再度入力を促してきます。ファイルが見つかると，画面には“....”が繰り返し表示されると思います。“....”が表示されている間はディスクからデータを読み取っていますので，絶対に電源を切ったりディスクをイジェクトさせないでください。最悪の場合ファイルが壊れてしまいます。

無事にファイルを最後まで読み込むと，たとえば図1のように表示されます。コメントエリアにはSV.Xで表示するときの画像変換の有無やアドバイスを表示します。

あ，それからプログラムは無限ループになってるので，終了するときはBREAKキーを押してください(手抜き)。

IFF，ILBMとは?

IFF(Interchange File Format)とはAMIGAで定められている共通フォーマットのことです。ILBM(InterLeved bitplane Bit-Map image)というのは画像関係のIFFであることを表します。

IFFファイルではさまざまな情報をチャンク(chunk)という構造体に格納しています。リストの170～250行が今回のプログラムで調べているチャンク名です。たとえば200行で定義してるBMHDチャンクには，画像のサイズや使用プレーン数などが格納されています。チャンクの情報が公開されているので，根性さえあればSV.Xのように他機種でAMIGAの画像を表示するビュアを作ることもできます。

図1　データ表示例

```
-----------------------------------
-----------------------------------
filename       :  badbird.ani
filetype       :  Animation File
bitplanes      :  352 x 440 x 6
colors         :  HAM(4096 colors)
display mode   :  Interlase
compression    :  Eric …(中略)… mode
total frame    :  606 frames
loop mode      :  ON
===================================
===================================
          コメントエリア
                :
                :
                :
```

どうなるSV.X

AMIGAの画像データをX68000にもってくるSV.X。本文中で述べているように，まだまだ発展途上の段階です。今回のプログラムのようにサポートも完全でない部分やデータによっては動かないものが存在しています。

また，せっかくですから配れる画像データをなるべく皆さんにも見せたいものです。しかし，アニメーションともなるとデータサイズは，結構大きくなるの場合があり，おいそれと配れるものではありません。

結局，AMIGAからデータをもってこれる環境のある人だけ，このSV.Xを有効利用できるのが現状です。せめて，X68000上でコンバートできるようになれば，かなり利用価値が上がるのですが……。誰か，ファイルコンバータを作ってくれないかなあ。

それでも，AMIGAは3.5インチ2DDでデータを供給しているから，コンバータができてもさらに3.5インチドライブが必要になるんですよね。いっそ，AMIGA用の増設ドライブをどうにかX68000に接続する方法を，考えたほうがいいかもしれない。

最後に

現在SV.Xの最新版はZ-MUSICのサポートBBSにアップされています。最新版では使用メモリを減らすことに成功したので，2Mバイトの機種で表示できなかったアニメーションも表示できると思います。

またRAMディスクも壊さないようになっています。アニメーションファイルもいくつかありますので，よかったら覗きにきてください。

<参考文献>

AMIGA ROM Kernel Reference Manual DEVICES, Addison Wesley

リスト

```
10 /*****************************************************
20 /* IFF ILBMファイルに含まれている情報を表示する
30 /*
40 /*     ver.0.50          Written by Shadow Mountain
50 /*****************************************************
60  error off
70 /*****************************************************
80 /*   初期設定
90 /*****************************************************
100 char buffer(60)
110 char bmhd_data(100)
120 char camg_data(10)
130 char anhd_data(50)
140 char find_last=0,dlta_num=0
150 str file
160 str filetype=""
170 str form_chunk="FORM"
180 str ilbm_chunk="ILBM"
190 str anim_chunk="ANIM"
200 str bmhd_chunk="BMHD"
210 str camg_chunk="CAMG"
220 str anhd_chunk="ANHD"
230 str dlta_chunk="DLTA"
240 str last_chunk="LAST"
250 str ansq_chunk="ANSQ"
260 int ai,ansq_len,length(1),er=0
270 width 96
280 /*****************************************************
290 /*   ファイルオープン
300 /*****************************************************
310 while 1
320 repeat
330    input "Input filename : ",file
340    ai=fopen(file,"r")
350    if ai<0 then print "ファイルはありません"
360 until ai<>-1
370 fread(buffer,8,ai)
380 if strcmp(form_chunk,0,3)=-1 then er=1:break
390 fread(buffer,4,ai)
400 if strcmp(ilbm_chunk,0,3)=0 then {
405    filetype="Picture File" } else {
410 if strcmp(anim_chunk,0,3)=0 then {
415    filetype="Animation File" } else {
420 er=2 }}
430 /*****************************************************
440 /* チャンクを読み込む
450 /*****************************************************
460 while feof(ai)<>-1
470   fread(buffer,4,ai)
480   if strcmp(form_chunk,0,3)=0 then {
490      fseek(ai,4,1) } else {
500      if strcmp(ilbm_chunk,0,3)=-1 then {
510         if strcmp(bmhd_chunk,0,3)=-1 then {
520            if strcmp(camg_chunk,0,3)=-1 then {
530               if strcmp(anhd_chunk,0,3)=-1 then {
540                  if strcmp(dlta_chunk,0,3)=-1 then {
550                     if strcmp(last_chunk,0,3)=-1 then {
560                        if strcmp(ansq_chunk,0,3)=-1 then {
570                        fread(length,1,ai)
580                        length(0)=even(length(0))
590                        fseek(ai,length(0),1) } else {
600                     fread(length,1,ai)
610                     length(0)=even(length(0))
620                     fseek(ai,length(0),1)
630                     ansq_len=length(0) }} else {
640                  fread(length,1,ai)
650                  length(0)=even(length(0))
660                  fseek(ai,length(0),1)
670                  find_last=1 }} else {
680               fread(length,1,ai)
690               length(0)=even(length(0))
700               fseek(ai,length(0),1)
710               dlta_num=dlta_num+1 }} else {
720            fread(length,1,ai)
730            length(0)=even(length(0))
740            fread(anhd_data,length(0),ai) }} else {
750         fread(length,1,ai)
760         length(0)=even(length(0))
770         fread(camg_data,length(0),ai) }} else {
780      fread(length,1,ai)
790      length(0)=even(length(0))
800      fread(bmhd_data,length(0),ai) }
810   }
820 }
830 endwhile
840 fclose(ai)
850 /*****************************************************
860 /*   画面表示
870 /*****************************************************
880 print
890 print"================================================="
900 print "filename        : ";file
910 print "filetype        : ";filetype
920 print "bitplane(s)     :";bmhd_data(0)*256+bmhd_data(1);
930 print "x";bmhd_data(2)*256+bmhd_data(3);"x";bmhd_data(8)
940 print "colors          :";
950 if (camg_data(2) and 8)=8 then {
960    print " HAM(4096 colors)" } else {
970    if camg_data(3) and &H80=&H80 then {
980       print " HALFBRITE(64 colors)" } else {
990       print pow(2,bmhd_data(8));"colors"
1000    }
1010 }
1020 print "display mode  : ";
1030 if camg_data(3) and 4=1 then {
1032    print "Lace" } else print "Interlase"
1040 if filetype="Animation File" then {
1050   print "compression   : ";
1060   switch anhd_data(0)
1070     case 0:print "set directry":break
1080     case 1:print "XOR ILBM mode":break
1090     case 2:print "Long Delta mode":break
1100     case 3:print "Short Delta mode":break
1110     case 4:print "Generalized short/long Delta mode":break
1120     case 5:print "Byte Vertical mode":break
1130     case 6:print "Stereo op 5":break
1140     case 74:print "Eric Grahams compression mode"
1150   endswitch
1160   print "total frames  :";
1170   if ansq_len=0 then {
1180     if find_last=0 then {
1182        print dlta_num-1; } else print dlta_num+1;
1190   } else {
1200     print ansq_len/3;
1210   }
1220   print "frames"
1230   print "loop mode     : ";
1240   if find_last=0 then print "ON" else print "OFF"
1250 }
1260 print"================================================="
1262 print
1270 print "SV．Xを使ってX68000で表示させる場合":print
1280 if (camg_data(2) and 8)=8 then {
1290    if bmhd_data(2)*256+bmhd_data(3)>256 then {
1300 print "4096色→256色にオーダードディザ法で変換，表示します"
1310 print "Qスイッチを指定すると使用頻度の多い256色を使って変換します"
1320 print "Qスイッチを指定したほうが変換後の画像が綺麗に仕上がります"
1330 print
1340    }
1350 print "水平型→垂直型変換が必要ですが，そのまま表示できます"
1360 }
1370 if bmhd_data(8)<=4 then {
1380 print "テキスト画面に表示できます" } else {
1390 print "Sスイッチを指定しておくと，水平型→垂直型";
1400 print "変換した結果をファイルに保存します"
1410 print "次回，同じファイルを表示する場合に，処理時間が大幅に短縮されます"
1420 }
1430 print
1440 endwhile
1450 switch er
1460   case 1:print "IFF形式のファイルではありません":break
1470   case 2:print "IFF ILBMまたはIFF ANIM形式のファイルではありません"
1480 endswitch
1490 end
1500 /*****************************************************
1510 /* 文字変数aと buffer(s)~buffer(e) の内容を比較する
1520 /*
1530 /* 同じなら0，違うなら－1を返す
1540 /*****************************************************
1550 func int strcmp(a;str,s;int,e;int)
1560   int  i=0,ret=0
1570   if pos>94 then {
1572      locate 0,csrlin:print space$(95);:locate 0,csrlin }
1580   print ".";
1590   while s<=e
1600     if a[i]<>buffer(s) then ret=-1:break
1610     i=i+1
1620     s=s+1
1630   endwhile
1640   return(ret)
1650 endfunc
1660 /*****************************************************
1670 /* 引き数を偶数にする
1680 /*****************************************************
1690 func int even(i;int)
1700    i=i+1
1710    i=i and &HFFFFFFFE
1720    return(i)
1730 endfunc
```

ファイルから文字列を探し出す

STRFIND.C

ショートプロぱーてぃでお馴染みの(で)氏はアルゴリズムで勝負
プログラムはファイル中の特定文字列をインデックスを使って検索するもの
足りない部分を補い，皆さんの手で立派なプログラムにしましょう

Komura Satoshi 古村 聡

概要と使い方

ライター陣によるショートプログラムということなので，私(で)は，ちょっとアルゴリズムに凝ったプログラムにしてみました。今回のプログラムはファイル中から特定の文字のある場所を探し出す，STRFIND.Cです。

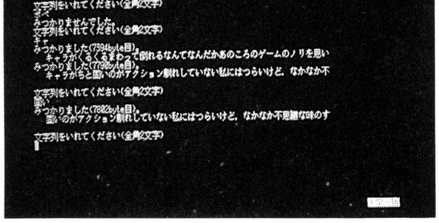
このプログラムは文字の位置を加工してインデックスとして確保しておき，テキストデータの検索の高速化に役立てる，いわゆるインデックスドテキストサーチという手法を用いたテキスト検索プログラムです。これがあれば，一度ファイルを読み込んでしまえば検索対象のファイルが100Kバイトあろうが1Mバイトあろうがテキストデータの中から特定の文字列(全角2文字)を高速に探し出してくれます。

プログラムはCで書かれていますので，エディタなどでリストを打ち込んだあと，

A>CC /Y /Gs1FFF STRFIND.C

としてコンパイルしてから使ってください。もちろんGCCがいいという人は，そちらでコンパイルしてもらっても問題ないはずです(その際スタックはできるだけ大きめに取っておいてください)。

さて，このコマンドを実行すると検索されるテキストデータのファイル名を聞いてきますのでそこでファイル名を入力するとファイルの読み込みを始めます。このときにプログラムはインデックスを作っています(このインデックスを作るのにそれなりに時間がかかります)。



インデックスが作り終わりましたらあとは検索するだけです。全角文字を2文字入れてください。結構高速に検索できるはずです。ただしその2文字は，

1文字目がひらがな，記号以外の文字
2文字目が記号以外の文字(ひらがな可)

となっています。

文字列を聞いてきたときに半角文字で，

++++

を入れるとこのプログラムを終了します。

プログラム解説

そうたいしたことはしていません(だからこそ200行以内に収まったのだ)。

このプログラムでは漢字やその送りの1文字など，テキストファイル中の検索対象になる文字のテーブルを作り，その文字の出てきた位置の情報を1つひとつ漢字の文字種類ごとにその文字の位置を確保しています。

たとえば「亜」という文字がファイル中に140,288,590,1009,2008,8134バイト目にあった場合，「亜」という文字のインデックスデータが，

亜　140,288,590,1009,2008,8134
唖　……
娃　………
　⋮
　⋮

というふうにメモリ上に作られます。

そして，検索するときには，たとえば「亜あ」という文字列を探したいときには「亜」の文字と「あ」という文字のインデックス情報を探し出し，

亜　140,288,590,1009,2008,8134
あ　133,235,592,1011,5005

そして「あ」の出現位置が「亜」の出現位置の次の文字(2バイト先)である「亜」の位置を調べるのです。

この例で手順をトレースしてみると，

・140バイト目の「亜」に対応する「あ」はない
・288バイト目も違う。
・592バイト目に「あ」がある（文字を見つけた)
・1009バイト目も……

というふうに見つけ出しているのです。

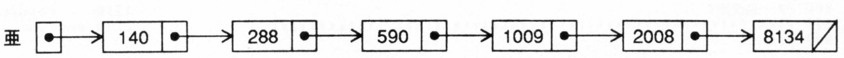
この方法はインデックスを作るのにかなり多くのメモリが必要で，文字種を示すおよそ27000個の配列+文字位置の格納エリアが検索対象出現文字数×8(long型×2で使っているので)個分のメモリを必要とします。そのため，ファイル中のすべての文字列を検索対象とすると，1Mバイトのファイルを扱うのに4Mバイトものワークエリアを必要としてしまうのです。

そのため，今回は先ほどいったように1文字目がひらがなだと使えない，というように制約を加えてあります。これだと，(経験的に)普通の文章は全文字中に占めるひらがなの割合が50%近くあるので，かなりメモリの節約が期待できるからです。

今回の文字種テーブルの領域を確保するのにメモリは，「普通」の文章を扱うことを前提に確保しています。そのためあまり漢字の多い文章(たとえば般若心経とか)を入れると，アドレスエラーになったり，終了したあとでCOMMAND.Xに戻れなくなる可能性があります(そのへんのチェックは全然していないので)。インデックス作成時に表示されるget cellがused cellの数より多くなっていないことを確認してください。

そうそう，文字テーブルに文字の出現位置を，と簡単にいっていますが，そこは単純な配列ではなく図1のようにリスト構造

図1　インデックスデータのデータ構造



になっています(まさか，文字が出てくるたびにREALLOCするわけにもいきませんのでね)。

最後に

短いわりには結構お得なプログラムなんではないか，などと作った本人は思ってしまっているのですが，実はまだ結構改良の余地があったりします。

たとえばこの検索プログラム，インデックスを作成するのにかなり時間がかかるんですが，ファイル読み込みを1バイトずつやっているので一気に読めばもう少し速くなるのではないかなとか，あるいは検索自体も同じ文字の出現回数が増えるとちょっと遅くなるんですが，2つの文字のテーブルの内容を比べるときに総当たりで比較してるのでこれを，うまくやればもうすこし比較の回数が減らせるのではないか，とか……(よく見るとやたらボロの出てくるプログラムかもしれない)。あ，あと，インデックスをいちいち作るのは時間がもったいないのでセーブ&ロード機能があれば，時間の節約にもなるしいいかもしれないですね。

今回発表したものは骨格のみの単純なモデルです。でもでも，せっかくない知恵しぼって作り出したアルゴルリズムなんで，よかったらしっかり改良して使ってやってください。

リスト

```
#include <stdio.h>
#include <stdlib.h>
#include <stat.h>
#include <jfctype.h>
#include <jstring.h>

struct ADDRINFO{
        struct ADDRINFO *pNext;
        unsigned long Addr;
}ADDRINFO;
struct ADDRINFO *MemArea;
typedef struct ADDRINFO *ADDR;
ADDR  IdxTable[27000];
ADDR  EndTable[27000];
int   AreaNo,Cont;

void InitIdxTable(.)
{
        int i;
        for(i=0;i<27000;i++)
                IdxTable[i]=NULL;
}

unsigned int ReadNext(int fd,int *Counter,int ReadBytes)
{
        unsigned char Code1,Code2;
        unsigned int  Code;

        do{
        long size;

        read(fd,&Code1,1);
        if(iskanji(Code1)){
          read(fd,&Code2,1);
          Code=(unsigned int)(Code1<<8)
              |(unsigned int)Code2;
                if(!jisspace(Code) && !jiskigou(Code)){
                 if(!jishira(Code)){
                    Cont=1;
                    /*putchar(Code1);putchar(Code2);*/
                    return Code;
                 }else{
                    if(Cont==1){
                    Cont = 0;
                    /*putchar(Code1);putchar(Code2);*/
                    return Code;
                        }
                }
                }
                (*Counter)+=2;
                if(*Counter >= ReadBytes) return 0;
        }else{
                ( *Counter )++; /*スキップする*/
                if ( *Counter >= ReadBytes) return 0;
                Cont = 0;
                }
        }while(1);
}

void RegistCode(unsigned int Code,int Counter)
{
    struct ADDRINFO *pCell;

    pCell=&MemArea[AreaNo];
    AreaNo++;
    if(pCell == NULL) printf("memory error¥n");
    if(IdxTable[Code] == NULL)  IdxTable[Code] = pCell;
    else                        EndTable[Code]->pNext = pCell;
    EndTable[Code] = pCell;
    pCell->pNext = NULL;
    pCell->Addr  = Counter;
}

int makidx(char *pFilename)
{
        int ReadBytes,CharCounter;
        int fd;
        struct stat FileStat;
        unsigned int Code;

        InitIdxTable();Cont=0;
        if( 0 != stat(pFilename, &FileStat) ) return 0;
        ReadBytes = FileStat.st_size;CharCounter = 0;
        MemArea=MALLOC2(2,(sizeof(struct ADDRINFO))*(Rea
dBytes/3));AreaNo=0;
        printf("%ld cell-areas get.¥n",ReadBytes/3);
        if( !(fd=open(pFilename,O_RDONLY|O_BINARY,0)))
                return 0;
        do{
           Code = ReadNext(fd,&CharCounter,ReadBytes);
           if(!Code)break;
           /*インデックスを作る*/
           RegistCode(Code-(unsigned int)0x8141,
                   CharCounter);
           CharCounter+=2;
        }while(CharCounter<ReadBytes);
        close(fd);
        printf("%ld cell used.¥n",AreaNo);
        return 1;
}

void FindWord(struct ADDRINFO *pA1,
                  struct ADDRINFO *pA2,
                  char pFilename[])
{
        int flag;
        struct ADDRINFO *pADDR1,*pADDR2;
        char StrBuf[256];

        flag =0;pADDR1=pA1;pADDR2=pA2;
        while(pADDR1 != NULL){
          pADDR2 = pA2;
          while(pADDR2 != NULL){
                if(pADDR1->Addr==((pADDR2->Addr)-2L)){
                unsigned char TmpChr;
                int fd,i;

                printf("見つかりました(%ldbyte目)。¥n"
                        ,pADDR1->Addr);
                fd = open(pFilename,
                           O_RDONLY|O_BINARY, 0 );
                lseek(fd,pADDR1->Addr,SEEK_SET);
                read(fd,StrBuf,74);
                for(i=0;i<75;i++)
                        if((StrBuf[i]==0x0d)
                           ||(StrBuf[i]==0x0A))
                              StrBuf[i] = 0x20;
                TmpChr=StrBuf[72];
                if(iskanji(TmpChr)) StrBuf[72]='¥0';
                else                StrBuf[73]='¥0';
                        printf("   %s¥n",StrBuf);
                flag = 1;
                close(fd);
                }
                pADDR2 = pADDR2->pNext;
          }
          pADDR1 = pADDR1->pNext;
        }
        if(flag == 0) printf("見つかりませんでした。");
}

main()
{
        unsigned char Dest[6],filename[255];
        unsigned int i;
        unsigned int Dest1,Dest2;

        printf("ファイル名を入れてください>");
        scanf("%s",filename);
        printf("¥n  *** インデックス作成中***¥n¥n");
        if( 0 == makidx( filename )){
           printf("インデックスが作れませんでした.");
           return -1;
        }
        while(1){
                printf("¥n文字列を入れてください(全角2文
字)¥n");
                scanf("%s",Dest);
                if(Dest[0] == '+') break;
                Dest1 = Dest[0] <<8 | Dest[1];
                Dest1 -= (unsigned int) 0x8141;
                Dest2 = Dest[2] <<8 | Dest[3];
                Dest2 -= (unsigned int) 0x8141;
                FindWord(IdxTable[Dest1],
                         IdxTable[Dest2],
                         filename);
        }
        MFREE(MemArea);
}
```

タイムトライアル3Dゲーム
MAGICAL TRIANGLE
(要MAGIC.X，MAGIC.FNC)

ショートプログラムとはいえ，ゲームなら派手で面白いものがいい
ここではMAGIC.FNCを使った3Dゲームを紹介しましょう
浜崎氏はどんなMAGICを見せてくれるのでしょうか

Hamazaki Masaya 浜崎 正哉

やっぱりMAGICよね

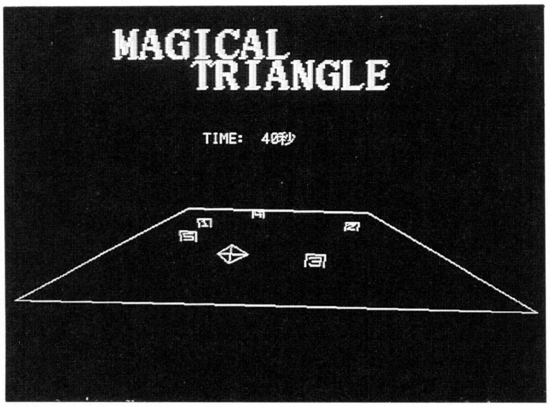
なんの因果関係があるのかわからないけど，Oh! X 5周年がめでたいのでショートプログラムを発表します。最近はMAGICづいている僕のことですから，今回もMAGICを使ってゲームを作ってみました。言語はもちろんX-BASIC。せっかく1991年7月号でMAGIC.FNCが発表されているんですから使わないテはありませんね。

ゲームは「三角錐を滑らせてゲートを取ってけゲーム」。このままタイトルにしようかと思ったけど，ちょっとナニだったので「MAGICAL TRIANGLE」という意味不明なタイトルをつけました。

遊び方

それでは遊び方を伝授しましょう。プログラムをRUNするとタイトル下にプレイ時間と平面を表すBOXに三角錐，そして数字の書かれた番号ゲート(?)が用意されます。プレイヤーは平面を傾け，数字の順番どおりゲートを通過するように三角錐を誘導していくことを目的とします。正しいゲートを通過するごとにそのゲートは消えていき，間違ったゲートに接触した場合は，ペナルティとして10秒加算されます。

全部のゲートを通過するとゲームコンプリート，タイムアタック制のわりにはベストタイムも表示されないシンプルさですが，ほしかったら自分でなんとかしましょう。そうそう，プログラムをRUNすると，いきなりゲームが始まりますので気をつけてください（手抜きありあり)。



操作にはマウスを使用し，上下に動かすことでPITCH方向の角度が変化し，左右に動かすことでBANK方向の角度が変化します。三角錐は傾いている方向へ徐々に加速していくようになっています。

また，ゲームの処理がちょっと重いため，マウスを転がすスピードを遅めにしないとうまく傾けることができないので，注意しましょう。

うだうだ説明するよりも実際に遊んでみてください。結構，難しいですよ。

プログラムについて

これといって特殊なことをしていないので（う～ん常套文句)，注釈行を参考にがんばって解析してみてください。

ただ，MAGICにオブジェクトを定義する関数MAGIC_CHRPUT()の部分は，MAGICのパラメータに関する知識が多少必要なので，軽く説明しておきます。

今回問題となったのは，平面の傾きと同じ傾きで平面上の物体を動かすところです。ここで回転の中心はMAGICの原点ではなく，平面の中心点になることに気づく必要があります。要するに，ただ平面の傾きだけ回転させ，オブジェクトの位置する座標まで移動するだけでは実現不可能なのです。実現するためには，回転中心座標をうまく使わなければなりません。

手順としては，回転中心座標を使いMAGICの原点を中心にして回転させます。そして，回転後平面の中心座標まで移動すればいいのです。具体例として，平面に中心座標が，H(100,100,100)，その平面上に乗っているオブジェクトの座標が，OBJ(50,100,70)とします。このオブジェクトを定義するときは，以下のパラメータを与えてやればいいことになります(HEAD,PITCH,BANKは同じなので省略)。

パラメータ	値
0	100
1	100
2	100
3	−50
4	0
5	−70

言葉だけではなかなか理解しにくい部分ですから，簡単なモデルを作り，いろいろ実験してみてください。忘れてならないことは，回転中心座標の座標系はMAGICの座標系と正負が逆である，そして計算順序が回転が行われたあと物体位置へ移動する，の2点です。

ここらへんの込み入った話は，1992年7月号の「MAGICとSIONIIの関係」で詳しく説明されているので，もっと知りたい方は参考にしてください。

プログラミングは続くよ……

ん～，久しぶりのショートプログラム。しかもBASICなんかで作ってしまった。あいかわらず物忘れが激しい僕だからすっかりBASICなんて忘れてましたね。ちょっとわからないことがあると，あっちゃこっちゃのスタッフを捕まえて質問攻めにしたり（少しは自分で考えろっての)，ずいぶん迷惑をかけたなあ。いつものとおり，といえばそうなんだけど。

でも，プログラミングっていいですね。あの完成したときの充実感。今回はショートプログラムながらもそれなりに充実感を得られたし，ちょっと気分転換にもなったかな。

てなわけで，気分転換もすんだことだし，制作途中のちょっとでっかいアセンブラソースと再び格闘することにしましょう。

リスト

```
10 /*
20 /* MAGIC.FNC用
30 /*      「三角錐を滑らせてゲートを取ってけゲーム」
40 /*                                 by M.Hamazaki
50 /*
60          int i,j,k,end_flag,over_flag
70          int box_obj,tri_obj,score
80          int ball_x,ball_y
90          int head,pitch,bank
100         int gate_max=4
110         int next_num,all_time,touch_flag
120         str befor_time
130         float a_x,a_y
140         dim int gate_xy(4,1)       /*登場するゲートの座標
150         dim int buf_p(4)           /*バッファポインタ格納
160         dim gate_flag(4)
170         dim int box_chr(21)
180         dim int tri_chr(25)
190         dim int ichi_chr(41),ni_chr(43)
200         dim int san_chr(56),yon_chr(41)
210         dim int go_chr(56)
220         dim int gate_data(15,1)
230 /*
240 /* オブジェクトデータ
250 /*
260 box_chr={4,200,0,200,-200,0,200,
270          -200,0,-200,200,0,-200,
280          +4,0,1,1,2,2,3,3,0}
290 tri_chr={4,0,-15,0,0,5,-20,
300          +17,5,8,-17,5,8,
310          +6,0,1,0,2,0,3,1,2,2,3,3,1}
320 ichi_chr={9,-10,-18,0,-10,0,0,10,-18,0,
330          +10,0,0,0,-14,0,-4,-12,0,0,-2,0,
340          -4,-2,0,4,-2,0,
350          +6,0,1,0,2,2,3,4,5,4,6,7,8}
360 ni_chr={9,-10,-18,0,-10,0,0,10,-18,0,
370           +10,0,0,6,-2,0,-4,-14,0,6,-14,0,
380           -6,-2,0,-6,-12,0,
390           +7,0,1,0,2,2,3,7,4,5,6,6,7,5,8}
400 san_chr={12,-10,-18,0,-10,0,0,10,-18,0,
410         +10,0,0,6,-2,0,-4,-14,0,6,-14,0,
420         -6,-12,0,-4,-2,0,-6,-4,0,
430         -2,-8,0,6,-8,0,
440         +9,0,1,0,2,2,3,5,6,6,4,5,7,4,8,8,9,10,11}
450 yon_chr={9,-10,-18,0,-10,0,0,10,-18,0,
460         +10,0,0,4,-2,0,4,-14,0,6,-8,0,
470         -6,-8,0,-6,-14,0,
480         +6,0,1,0,2,2,3,5,4,6,7,7,8}
490 go_chr={10,-10,-18,0,-10,0,0,10,-18,0,
500         +10,0,0,6,-2,0,6,-8,0,-6,-8,0,
510         -6,-14,0,6,-14,0,-6,-2,0,
520         +8,0,1,0,2,2,3,5,4,5,6,6,7,7,8,9,4}
530 /*
540 /* メインルーチン
550 /*
560         all_init()
570         while end_flag = 0
580          game_init()
590           while next_num <> 5
600             TIME_COUNT()
610              MOUSE_IN()
620             BALL_MAIN()
630             TOUCH_CHK()
640             MAGIC_CHRPUT()
650           endwhile
660          TRY_AGAIN()
670         endwhile
680         end
690 /*
700 /* 初期化
710 /*
720 func all_init()
730      int i,j,k,sr,sm
740         magic_flush()            /*データバッファのクリア
750         magic_init()             /*3Dワークの初期化
760         magic_screen(4)          /*画面モード512×512
770         for i=0 to 8             /*3Dパラメータクリア
780            magic_para(i,0)
790         next
800         magic_color(15)             /*描画色のセット
810 /*オブジェクトをバッファに放り込む
820         box_obj=magic_putbuf(2,box_chr)
830         tri_obj=magic_putbuf(1,tri_chr)
840         buf_p(0)=magic_putbuf(3,ichi_chr)
850         buf_p(1)=magic_putbuf(3,ni_chr)
860         buf_p(2)=magic_putbuf(3,san_chr)
870         buf_p(3)=magic_putbuf(3,yon_chr)
880         buf_p(4)=magic_putbuf(3,go_chr)
890           mouse(4)               /*マウスの初期化
900           mouse(2)               /*マウスカーソルの消去
910         screen 1,1,1,1           *グラフィックモードを合わせる
920         console ,,0              /*ファンクションキーの表示OFF
930         apage(2)
940         symbol(98,22,"MAGICAL",2,2,2,15,0)
950         symbol(100,20,"MAGICAL",2,2,2,11,0)
960         symbol(98,62,"   TRIANGLE",2,2,2,15,0)
970         symbol(100,60,"   TRIANGLE",2,2,2,11,0)
980         for j=0 to 3              /*座標初期データセット
990          for k=0 to 3
1000           gate_data(j*4+k,0)=k*80-120
1010           gate_data(j*4+k,1)=j*80-120
1020         next
1030        next
1040 endfunc
1050 /*
1060 /* ゲームの初期化
1070 /*
1080 func game_init()
1090      int rnd_1,rnd_2,work
1100      int i,j,k
1110         cls
1120         head=0:pitch=0:bank=0
1130         ball_x=180:ball_y=-180
1140         a_x=0:a_y=0
1150         next_num=0:score=0
1160         over_flag=0:end_flag=0
1170         for j=0 to 20            /*座標をシャッフル
1180           rnd_1=int(100*rnd() mod 16)
1190           rnd_2=int(100*rnd() mod 16)
1200              work=gate_data(rnd_2,0)
1210              gate_data(rnd_2,0)=gate_data(rnd_1,0)
1220              gate_data(rnd_1,0)=work
1230              work=gate_data(rnd_2,1)
1240              gate_data(rnd_2,1)=gate_data(rnd_1,1)
1250              gate_data(rnd_1,1)=work
1260         next
1270         for i=0 to gate_max /*ゲート出現位置初期化
1280           gate_xy(i,0)=gate_data(i,0)
1290           gate_xy(i,1)=gate_data(i,1)
1300           gate_flag(i)=1
1310         next
1320         MAGIC_CHRPUT()
1330         locate 23,10:print"TIME:"
1340         all_time=0
1350         befor_time=time$
1360 endfunc
1370 /*
1380 /* TRY AGAIN?
1390 /*
1400 func TRY_AGAIN()
1410    int chk=0
1420    str k
1430         symbol(150,240,"TRY AGAIN(Y/N)?",1,1,2,9,0)
1440         while chk=0
1450           k=inkey$(0)
1460           if k="y" or k="Y" then {
1470              end_flag=0:chk=1 }
1480           if k="n" or k="N" then {
1490              end_flag=1:chk=1 }
1500         endwhile
1510         symbol(150,240,"TRY AGAIN(Y/N)?",1,1,2,0,0)
1520 endfunc
1530 /*
1540 /* 時間カウント
1550 /*
1560 func TIME_COUNT()
1570         if befor_time <> time$ then {
1580            all_time=all_time+1:befor_time=time$ }
1590      locate 28,10:print using"####";all_time;
1600      print "秒"
1610 endfunc
1620 /*
1630 /* 当たり判定
1640 /*
1650 func TOUCH_CHK()
1660     int kosu=0
1670         for i=0 to 4
1680          if gate_flag(i) <> 0 then {
1690             dx=abs(ball_x-gate_xy(i,0))
1700             if dx<20 then {
1710               dy=abs(ball_y-gate_xy(i,1))
1720               if dy<20 then {
1730                 if i=next_num then {
1740                   gate_flag(i)=0
1750                   beep
1760                   next_num=next_num+1
1770                 } else {
1780                         kosu=kosu+1
1790                         if touch_flag = 0 then {
1800                            all_time=all_time+10
1810                            touch_flag=1:beep }
1820          }  } } }
1830         next
1840         if kosu = 0 then touch_flag=0
1850 endfunc
1860 /*
1870 /* 玉がすべる
1880 /*
1890 func BALL_MAIN()
1900         a_x=a_x-6.672#*sin(bank*pi()/180)
1910           if a_x >  10# then a_x =  10
1920           if a_x < -10# then a_x = -10
1930          ball_x=ball_x+int(a_x)
1940         a_y=a_y+6.672#*sin(pitch*pi()/180)
1950           if a_y >  10# then a_y =  10
1960           if a_y < -10# then a_y = -10
1970          ball_y=ball_y+int(a_y)
1980         if ball_x >  200 then ball_x =  200
1990         if ball_x < -200 then ball_x = -200
2000         if ball_y >  200 then ball_y =  200
2010         if ball_y < -200 then ball_y = -200
2020 endfunc
2030 /*
2040 /* マウスで,PITCH(上下),BANK(左右)を変える
2050 /*
2060 func MOUSE_IN()
2070         int mx,my,bl,br,n
2080         msstat(mx,my,bl,br)
2090         pitch=pitch - my
2100           if pitch >  20 then pitch= 20
2110           if pitch < -20 then pitch=-20
2120         bank=bank - mx
2130           if bank >  20 then bank= 20
2140           if bank < -20 then bank=-20
2150 endfunc
2160 /*
2170 /* キャラクターを定義する
2180 /*
2190 func MAGIC_CHRPUT()
2200         magic_seek(2,box_obj,0)          /*BOXの定義
2210         magic_data(2)
2220         magic_para(0,0)                  /*XYZ座標
2230         magic_para(1,60)
2240         magic_para(2,350)
2250         magic_para(3,0)                  /*回転中心座標
2260         magic_para(4,0)
2270         magic_para(5,0)
2280         magic_para(6,head)               /*回転角
2290         magic_para(7,pitch)
2300         magic_para(8,bank)
2310         magic_pers()
2320         magic_seek(1,tri_obj,0)          /*三角錐の定義
2330         magic_data(1)
2340         magic_para(0,0)
2350         magic_para(1,55)
2360         magic_para(2,350)
2370         magic_para(3,-ball_x)
2380         magic_para(4,0)
2390         magic_para(5,-ball_y)
2400         magic_pers()
2410         magic_para(1,60)
2420          for i=0 to gate_max             /*ゲートの定義
2430          if gate_flag(i)<>0 then {
2440           magic_seek(3,buf_p(i),0)
2450           magic_data(3)
2460           magic_para(3,-gate_xy(i,0))
2470           magic_para(4,0)
2480           magic_para(5,-gate_xy(i,1))
2490           magic_pers() }
2500          next
2510         magic_disp()
2520 endfunc
```

乱数を使った

地形表示プログラム

2D，3Dを問わず描画が難しい自然の風景
ここでは独自のアルゴリズムで地形風データ作成をしてみた
地形図風，風雲のグラフィックが簡単に作成できるのだ。

Nakano Shuichi 中野 修一

ショートプログラムということで，最初はスキャナから取り込んだデータから7階調でSX-WINDOW用のアイコンを作成するプログラム（昔作ったもの）を出そうと思っていたら「それはショートプロぱーてぃでやるからダメですよ」と古村君にクギを刺されてしまった。次に簡単なRPGを作ろうとしたのだが，どうもショートプログラムには収まりそうにない。ということで，地形表示プログラムを作ることにした。

＊　＊　＊

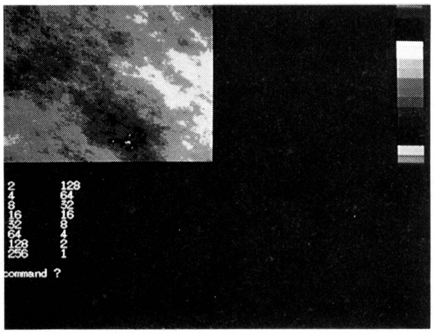
考えてみると日頃なにかほしいプログラムがあっても自分で作ることはほとんどない。ほかのスタッフをつかまえて，これこれこういうアルゴリズムで，こういう処理をして……云々と口説き落とす。「ほしいプログラム」というのが私の技術レベルを超えていることが問題なのだが，まわりにかなり上級のC言語プログラマかアセンブラ使いがいる環境のためでもある。

ということで，私が作るプログラムのほとんどは必然的にX-BASICのショートプログラムとなる。X-BASICならC言語と大差がないのでC言語くらい使えば……と思う人もいるだろう。実は私はC言語が嫌いな人だったのだ。加えて，同じ内容ならたいていの人よりは短いプログラムを作るからショートプログラムになってしまうのだ（貧乏性だなぁ）。

今回作った地形表示についても，いろいろ誘いをかけてみたのだが，どうもうまくない。X-BASICでグラフィックを扱うと速度的な限界が露骨に出てくるので自分ではやりたくなかったのだがしかたない。やれやれ。

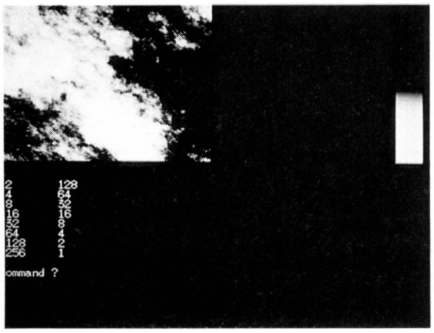
アルゴリズム

さて，地形表示の際に考慮したアルゴリズムは3つ。ひとつはZ's-EXのランダムフラクタルを使うもの。RF.DATを流用すれば処理が軽減できるかもしれない。次に3Dフラクタルを使うもの。だいたいの考え方はわかるので，まあ作れないことはないだろう。最後にAMIGAのFRSでよく使用されているChris Gray氏のアルゴリズムを使うこと。高速で非常に優秀である。

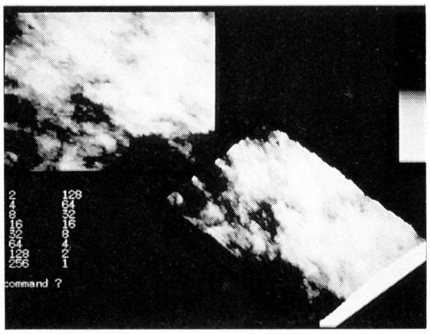


地形を作成

パレットで高さを変える



雲を作ることもできる



擬似3D表示だ

作成できる地形を多彩にしたいのでRF.DAT流用はパス。

Chris Gray氏のアルゴリズムはゲームなどで自由に使っていいとドキュメントに書かれているが，利用料をよこせとなっているのでなんとなくパス。そのうち紹介することもあるだろう。

とりあえず3Dフラクタルの線でいくことにするが，そもそも地形が自己相似形であるという考え方には疑問がある。某ゲームでのフラクタルによる「自然な海岸線」というのもとっても不自然にしか見えない。フラクタルでも地形っぽいものを作れることは実証されているが，だから地形＝フラクタルと思い込むのは危なそうだ。

結局，3Dフラクタルもパス。ここではあえてすべて乱数で地形を作成するアルゴリズムを作ってみた。基本描画過程は3Dフラクタルとほぼ同じで「自己相似」という部分をまったく無視したものである。普通にランダムフラクタルといわれているやつかもしれないが，詳しくは知らない。

まず描画エリアを4分割し，それぞれの領域をそこにあった色に乱数を重ねた色で塗り換える。処理が一巡したらエリアをさらに4分割して1ピクセルになるまで繰り返す。これが基本。実際には「そこにあった色」の位置にゆらぎを入れ，乱数の強度にもフィルタが入っている。こういうふうに徐々に積み重ねていくようにすれば，極端に飛び出したところはなくなるし，位置のゆらぎが単調な図形になることを防いでくれる。一部平均値を取っているところもあるので乱数でも不自然にはなるまい。

こういうプログラムは再帰を使うのが常道である。しかし再帰を使うと表示中の画面があまり面白くない。そこで再帰ループをFORループに変換してある。画面を介し

て値の受け渡しをやっているので，再帰にする必要はないのだ。本当のところをいえば，いちいち画面を塗らなくても計算できるのだけど，見た目を重視した。

今後の拡張

最初はまともなものが表示できるか自信がなかったのだが，なんとなくそれっぽいものは表示できるようになった。ちょっとうれしい。ついでなので，付加機能をいろいろ加えてみた。セーブもつけた。PICだと10行くらい増えるのでGL3にした。

一応，地形風と雲風のパレットを用意したが，各自の好みでパレットはいろいろ変えてみてほしい。作成された地形はまだ肌理が粗い感じなので改良の余地はある。3D表示しながら調整しないと難しい部分もありそうだ。

もともとZ's-EX用のプログラムとして考えていたものなので，雲はともかく，ちゃんとした3D表示が必須なのだが対応していない。最終的には地形と空の3D表示を行って，FLY TROUGHのアニメーションファイルを作成して……（Scenery Animaterだな），というところまで持っていきたいものである。

だいたいどのあたりをいじるといいかはわかっているので，高速化を含めて今後，改良されて再発表されるかもしれない。そのときこそは誰かほかの人に頼もう。

ま，とりあえずショートプログラムとしてはこんなところだろう。行数もきっちり128行になったし。

なお，使用できるコマンドは，

P　疑似3D表示
4　パレット変更（下）
6　パレット変更（上）
RETパレットリバース
2　パレットを雲風に
8　パレットを地形風に
N　次の地形を作る
S　セーブ（GL3形式）
Q　終了

となっている。

リスト1

```
10 int a,b,c,d,i,j,k,x,y,c0,size=256,com,pk,rev
20 int p1(15),p2(15,2)
30 str dm
40 int at1(15)={15,35,42,63,79,95,111,127
50              ,143,159,175,191,207,223,239,255}:/*地形用
60 int cl1(15)={15,75,80,103,135,138,140,142
70              ,145,148,153,159,167,180,199,255}:/*雲用
80 int at2(15,2)={130,30,15, 130,30,20,  20, 5,30,  37,28,25
90                , 44,21,23,  44,21,21,  55,20,19,  55,20,18
100               , 58,30,16,  67,24,12,  70,20,11,  80,20, 9
110               , 86,15, 7,  16,20,20,  16,30,15,  16,30,10}
120 int cl2(15,2)={125,30,20, 125,30,20, 125,30,20, 125,30,20
130               ,122,29,22, 116,29,22, 112,23,28, 106,20,30
140               ,106,15,30, 107,11,30, 110, 8,30, 110, 5,30
150               ,110, 4,31, 110, 2,31, 110, 1,31, 110, 0,31}
160 screen 1,2,1,1 :console 0,31,0
170 atpal()
180 for i=0 to 255:line(480,i,511,i,i):next
190 repeat  :/*メインルーチン-----------------------------------
200   cls
210   fill(0,0,size-1,size-1,150)
220   a=1
230   locate 0,18
240   repeat              :/*描画部メインループ
250     a=a*2:b=size/a
260       for i=0 to a-1
270       for j=0 to a-1
280         map()
290       next
300     next
310     print a,b
320   until b<3  :/*本当は2  デバッグ時は9
330   print:print"command ?"
340   repeat              :/*制御部メインループ
350     com=asc(inkey$(0))
360     switch com
370       case 's'  :gl3save():break
380       case 'p'  :pers():break
390       case '4'
400       case &H1D :pk=pk-2:if pk<-256 then pk=-256
410       case '6'
420       case &H1C :pk=pk+1:palset(pk):break
430       case '2'
440       case &H1F :atpal():break
450       case '8'
460       case &H1E :clpal():break
470       case 13 :rev=(rev=0):palset(pk)
480     endswitch
490     if com='q' then break
500   until com='n'
510 until com='q'
520 end :/*===================================================
530 func map()          :/*地形作成ルーチン---------------------
540 int x0,y0,x1,y1
550   x=j*b:y=i*b
560   k=(sqr(b/1.5#)*2)+1
570   sc=rnd()*k+20-k/2
580   x0=rnd()*b*2+x-b:if x0<0 then x0=0
590   x1=rnd()*b*2+x-b:if x1<0 then x1=0
600   y0=rnd()*b*2+y-b:if y0<0 then y0=0
610   y1=rnd()*b*2+y-b:if y1<0 then y1=0
620   if x0>size then x0=size-1
630   if x1>size then x1=size-1
640   if y0>size then y0=size-1
650   if y1>size then y1=size-1
660   c=(point(x0,y0)*sc+10)/20
670   c=c+(point(x1,y1)*sc+10)/20
680   if c<1   then c=1
690   if c>511 then c=511
700   c=(point(x,y)+c+1)/3
710   fill(x,y,x+b-1,y+b-1,c)
720 endfunc
730 func pers()         :/*エセ3D表示ルーチン-----------------
740 int ii,jj
750   for i=0 to size-1
760     ii=i/2
770     for j=0 to size*3/4-1
780       a=point(j*4/3,i  )
790       line(320+j-ii,210+j+ii,320+j-ii,210+j+ii-a/4,a)
800     next
810     line(320+j-ii,210+j+ii,320+j-ii,210+j+ii-a/4+1,230)
820   next
830   i=size-1:ii=i/2
840   for j=0 to size*3/4-1
850     a=point(j*4/3,i  )
860     line(320+j-ii,210+j+ii,320+j-ii,210+j+ii-a/4+1,250)
870   next
880 endfunc
890 func palset(k)     :/*パレット設定メイン関数--------------
900 int p,i,j,m
910   p=1
920   for i=0 to 15
930     for j=p to p1(i)
940       m=(j+k+256)mod 256
950       if rev<>0 then m=256-m
960       if m<>0 then palet(m,hsv(p2(i,0),p2(i,1),p2(i,2)))
970     next
980     p=p1(i)+1
990   next
1000   for i=0 to 63 :dm=inkey$(0):next
1010 endfunc
1020 func atpal()       :/*地形用パレット設定------------------
1030   for i=0 to 15:p1(i)=at1(i):next
1040   for i=0 to 15:for j=0 to 2:p2(i,j)=at2(i,j):next:next
1050   pk=0:palset(0)
1060 endfunc
1070 func clpal()       :/*雲用パレット設定--------------------
1080   for i=0 to 15:p1(i)=cl1(i):next
1090   for i=0 to 15:for j=0 to 2:p2(i,j)=cl2(i,j):next:next
1100   pk=0:palset(0)
1110 endfunc
1120 func gl3save()     :/*グラフィックセーブルーチン----------
1130 int buf(256),i,j,fn,c1,pp,c2
1140   fn=fopen("maptest.gl3","c")
1150   for i=0 to 255
1160     for j=0 to 255
1170       c1=(point(j,i)+pk+256)mod 256:pp=-1
1180       if rev<>0 then c1=256-c1
1190       repeat:pp=pp+1:until p1(pp)>=c1
1200       c2=hsv(p2(pp,0),p2(pp,1),p2(pp,2))
1210       buf(j)=(c2 shl 16)+c2
1220     next
1230     fwrite(buf,256,fn):fwrite(buf,256,fn)
1240     locate 55,1:print i*2
1250   next
1260   locate 55,1:print"       "
1270   fclose(fn)
1280 endfunc
```

最近流行の不思議図形

裸眼立体視(ランダムドット風)

Tan Akihiko　丹 明彦

裸眼立体視のなかでも見るのが難しいランダムドット法
砂目のような模様のなかから，不思議なくらい鮮明な立体映像に変わります
最新のアルゴリズム，ランダムドットにかなり近い立体図形作成プログラムです

先月の「大人のためのX68000」を受けて，ランダムドットによる立体視図形作成プログラムをお送りしよう。これがまた単純なアルゴリズムで，ショートプロにぴったりなのだ。

＊　＊　＊

プログラムは2本立て。1本めのdraw3d.xは立体視するための元絵を描くプログラム。768×512ドット16色のグラフィックで，パレットコードは高さを表す。パレットコード0の部分はもっとも低く，15の部分はもっとも高い。ここで描くのは中空の角錘の頭を切り落としたような形。引数に底面の多角形の角数を与えることができる。たとえば「draw3d 6」とやると6角形をベースにした図形を描く。

特にdraw3d.xを使わなくても，ペイントツールで描いたものでも構わない。さらに，自分でレンダリングのプログラムを書ける人なら，もっと凝ったことも可能だ。この絵はZバッファそのものだし，レイトレーシングの視点からの距離を画像化したものも使えそうだ。

これを2本めのrandomdot.xでランダムドットに変換する。テキスト画面にランダムドットパターンを描き出し，キー入力待ちの状態になる。画面の上で立体視することができるが，ここですかさずCOPYキーまたはSHIFT＋COPYキーを押すとハードコピーが取れる。経験的に紙に印刷したほうが立体視しやすい。裸眼立体視(特にランダムドット)は結構目に負担をかけるし集中力を要するので，チラチラするCRTは不向きなのだ。

引数に絵の名前を与えると，画面左上に表示する。「randomdot nyoronyoro」などのように使う。これがないと，ランダムドットはぱっと見には単なるランダムな点の集まりなので，なんの絵だかわからなくなってしまうのだ。そのためのタイトルを指定できるということ。

立体視の方法

なんだか世間では裸眼立体視がはやっているらしいが，立体視そのものは大昔からあった。その原理は単純で，要するに人間がものを見るときに生じる「視差」をシミュレートしてやろう，ということだ。右目用の画像と左目用の画像を用意し，右目には右目用の画像を，左目には左目用の画像を見せる。これが立体視のすべてだ。右目用の画像と左目用の画像は微妙にずれており，このずれが脳をダマして平面を立体に見せるのである。

裸眼立体視はそのなかでもかなり技術を要するもので(費用は要しないが)，右目用と左目用の2枚の絵を並べ，それを特別な器具なしで見るものだ。人間は普段，視線をひとつのものに集中するように見るものなので，右目と左目とで別のものを見るというのはけっこう大変なのである。

そしてランダムドットは，絵が1枚しかない。それでどうして立体視できるのか不思議なのだが，そこにはちゃんとカラクリがある。

とりあえず見方を説明すると，絵の上の2つの印を手がかりにして左右の目を裸眼立体視用にコントロールし，そこでおもむろに下のランダムパターンを見ると，あら不思議，そこには立体が浮かび上がっているというわけ。

では，いったいどのようにして1枚の絵で左右の目に別々の画像を提供しているのだろうか？

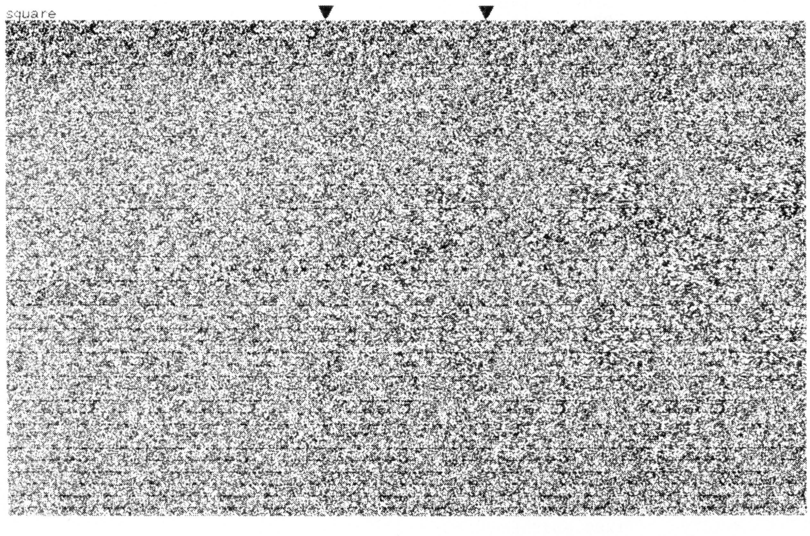
アルゴリズム

ランダムドット作成のアルゴリズムは単純(しかし思いつくのは難しい)。1枚の絵のように見えて，実は2枚の絵が巧妙に入っているのである。その2枚の絵を置く間隔をwドットとする。wは絵の幅の5分の1を目安にする(今回のプログラムでは絵の幅が750ドット，したがってw＝150)。手がかりとなる2つの印もこのwドットの間隔で配置する。

これでwドット幅の領域を5周期分描くと全体が埋まることになる。では，どのように描くか。

1周期め，絵の左端からwドットは，な

図1　出力例（四角形）



にも考えずにランダムに点を打つ。これが基本的なパターンになる。

残りの4周期がキモである。1周期前の絵を左目用の画像と見立てて右目用の画像を描くのである。右目画像と左目画像のずれは目からの距離で決まる。目に近いほど大きくずれる。

元絵(今回は16色のグラフィック)を見て注目する点の紙面からの高さを求める。高さが0(紙面上)なら1周期前のドットをそのままコピーする。高さがあれば，1周期前の場所からその高さに比例するドット数だけ右のドットをコピーする。こうすることで，高さに比例してずれる右目用の画像を作る。

1周期めの左目画像
2周期めの左目画像＝1周期めの右目画像
3周期めの左目画像＝2周期めの右目画像
4周期めの左目画像＝3周期めの右目画像
4周期めの右目画像

という関係になっているので，右目画像と左目画像を1枚の絵に押し込んでも破綻しないのである。このカラクリを知ったときは結構感動した。

＊　＊　＊

このランダムドットのアルゴリズムのヒントは海の向こうのコンピュータネットワークであるUSENETの電子ニュースから得た(ニュースグループ"alt.3d")。探せばちゃんとしたランダムドットのプログラムも流れていたはずだが，運悪く見つからなかったので，似たようなことをしている記事をもとにしてプログラムを書いてみたものがこれである。

今回紹介したランダムドットの元絵は簡単な模様でつまらないが(ショートプロという制約のせいにしておこう)，3次元の元絵を描画するプログラムとランダムドット変換プログラムとは分けてあるので，好きな絵を用意して立体視を楽しんでいただきたい。ただし，うまく立体に見える絵を描くのには多少コツが必要だ。

リスト1

```
/*
 *  draw3d.c
 *      ・768×512ドット16色グラフィック画面に3次元
 *      画像(パレットコードが高さを表す)を描画する。
 *      パレットコード0～15、値はzを表す。
 */

#include        <basic0.h>
#include        <graph.h>
#include        <math.h>
#include        <stdlib.h>

void    poly( int x, int y, int r, int n, int c )
{
    int i, x1, y1, x2, y2;
    x1 = x; y1 = y-r;
    for ( i = 1; i <= n; i++ ) {
        x2 = x - r*sin(PI*2.0*i/n);
        y2 = y - r*cos(PI*2.0*i/n);
        line( x1, y1, x2, y2, c, 0xFFFF );
        x1 = x2; y1 = y2;
    }
}

void    main( int argc, char *argv[] )
{
    int i, h, n;
    screen( 2, 0, 1, 1 );
    wipe();
    if ( argc < 2 ) n = 4; else n = atoi( argv[1] );
    for ( i = 200; i > 0; i-- ) {
        h = 8 + 7.5*cos(PI*1.0*(i-100.0)/100.0);
        poly( 450, 264, i+50, n, h );
        paint( 450, 264, h );
    }
}
```

リスト2

```
/*
 *  randomdot.c      1992/10/24       A.Tan
 *      ・768×512ドット16色グラフィック画面に描いた
 *      3次元画像(パレットコードが高さを表す)を裸眼
 *      立体視用のランダムドットパターンに変換して
 *      テキスト画面に描画する。
 *      ・上16ドットと右18ドットは使わない。
 *      ・(SHIFT+)COPYキーでハードコピーを取れる。
 *      ・コンパイル:
 *      gcc randomdot.c -O -lbas -liocs -lfloatfnc
 */

#include        <stdio.h>
#include        <conio.h>
#include        <stdlib.h>
#include        <graph.h>
#include        <iocslib.h>

void    drawNotch()     /* 裸眼立体視のための目印 */
{
        struct FNTBUF   fb;
        /* "▼"のフォントパターン */
        FNTGET( 8, 0x81A5, &fb );
        /* プレーン0 */
        TCOLOR( 1 );
        TEXTPUT( 300-8, 0, &fb );
        TEXTPUT( 450-8, 0, &fb );
        /* プレーン1 */
        TCOLOR( 2 );
        TEXTPUT( 300-8, 0, &fb );
        TEXTPUT( 450-8, 0, &fb );
        /* 元に戻す */
        TCOLOR( 3 );
}

unsigned char   gBuf[768/2], zBuf[768];
unsigned char   dBuf[768], tBuf[768/8];

void    drawRandomDot() /* ランダムドットで3次元化する */
{
    int i, j;
    for ( i = 16; i < 512; i++ ) {
        /* 取り込む */
        get( 0, i, 750, i, gBuf, 750/2 );
        /* zBufに展開 */
        for ( j = 0; j < (750/2); j++ ) {
            zBuf[j*2  ] = gBuf[j]/16;
            zBuf[j*2+1] = gBuf[j]%16;
        }
        /* 左端はただの乱数 */
        for ( j = 0; j < 150; j++ ) dBuf[j] = ((rand()/37)%2);
        /* 3次元化 */
        for ( j = 150; j < 750; j++ ) dBuf[j] = dBuf[j-150+zBuf[j]];
        /* 右端は処理しない */
        for ( j = 750; j < 768; j++ ) dBuf[j] = 0;
        /* tBufに展開 */
        for ( j = 0; j < 768; j += 8 )
            tBuf[j/8] = ((dBuf[j]<<7)|(dBuf[j+1]<<6)|
                        (dBuf[j+2]<<5)|(dBuf[j+3]<<4)|
                        (dBuf[j+4]<<3)|(dBuf[j+5]<<2)|
                        (dBuf[j+6]<<1)|(dBuf[j+7]));
        B_MEMSET( (unsigned char*)(0xE00000+i*128), tBuf, 768/8 );
        B_MEMSET( (unsigned char*)(0xE20000+i*128), tBuf, 768/8 );
    }
}

void    main( int argc, char *argv[] )
{
    int i;
    /* グラフィックを見えなくする */
    for ( i = 0; i < 16; i++ ) palet( i, 0 );
    /* テキスト画面全部を使う */
    printf( "¥x1B[>5h" );          /* カーソル非表示 */
    printf( "¥x1B[>1h" );          /* 最下行を確保する */
    printf( "¥x1B[2J" );           /* 画面消去 */
    /* タイトルを表示する */
    if ( argc < 2 ) printf( "Untitled¥n" );
            else    printf( argv[1] );
    /* 目印を描く */
    drawNotch();
    /* ランダムドットパターンを描く */
    drawRandomDot();
    /* キー入力を待つ */
    getch();
    /* テキスト画面を戻す */
    printf( "¥x1B[2J" );           /* 画面消去 */
    printf( "¥x1B[>1l" );          /* 最下行をHumanに返す */
    printf( "¥x1B[>5l" );          /* カーソル表示 */
}
```

対戦型シューティング

撃ち合いゲーム

まず着地，そして弾薬を補給したら戦闘開始！
慣性のついた自機をうまく操って敵を撃ち落とすのだ
人間対人間で遊ぶものなので，まず対戦相手の確保からはじめてください

Ishigami Tatsuya 石上 達也

ショートプログラムを1本だそうです。で，やっぱり，ショートプロなのですから，プログラムは小さくなければいけないのでしょう，きっと。プログラムを小さくするということは，複雑なルールを組み込めないわけです。どうやって面白くするかというと，組み込むのをあきらめて外から調達するのです。

普通，こういうゲームを「対戦ゲーム」といいます。

図1　キー操作

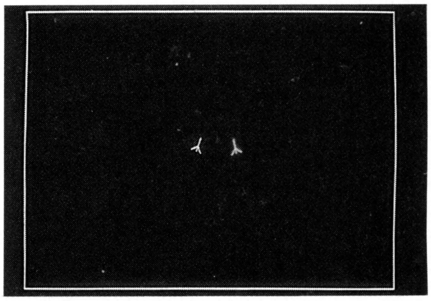
プレイヤー2	プレイヤー1
アクセル W	アクセル 8
A 左旋回 ← → D 右旋回	4 左旋回 ← → 6 右旋回
S ブレーキ	5 ブレーキ
CTRL　ミサイル発射	→ミサイル発射

遊び方

まず，自分のほかに暇そうにしている友人を1名ほど用意します。なるべく，どうでもいい友人のほうが好都合です。その友人とジャンケンをして，負けたほうがリスト1のプログラムを打ち込みます。打ち込みが終了してコンパイルが無事済んだら，いよいよプログラムを走らせて，ゲームスタートです。

友人をX68000のやや左側に座らせプレイヤー2をさせます。ということは，X68000のやや右側に座ったあなたはプレイヤー1です。で，操作方法は図1です。

さて，撃ち合いを始める前に弾を装着しなければなりません。これは，地上にうまく着地することにより10発装填されます。着地は，機首を上に向けて速度を抑えて行わなければなりません。作った私がいうのもなんですが，これはかなり難しい作業です。もちろん，ホバリング（?）しながらゆっくり着地してもよいのですが，相手が急降下&逆噴射などというテクニックを使った日には，イニシアチブを相手に取られてしまいます。とにかく，先に着地し，弾を装填したほうが有利です。



弾の装填がほぼ同じ時間なら，次は運転テクニックの勝負です。慣性&重力のきくゲームですから，早め早めの状況判断が肝心です。

さあ，友人をコテンパンにやっつけ，次のプログラムを入力させましょう。

リスト1

```
/*
  対戦型シューティングゲーム
  Programmed By T.Ishigami
  '92/10/21
*/

#include  <stdio.h>
#include  <basic.h>
#include  <graph.h>

#define   FALSE   0
#define   TRUE    ~FALSE

#define   WIDTH   768
#define   HIGHT   512
#define   PI      3.1415926
#define   MAX_CANON       10
#define   CANON_SPEED     0.04

typedef struct {
  int     x,y;
  int     velx,vely;
  int     rad;
  int     triger;
  int     count;
} CHDATA;

typedef struct {
  int     exist;
  int     x,y;
  int     rad;
} CANON;

int       sinT[256];        /* 三角関数テーブル */
int       cosT[256];

CHDATA    chdata[2];
CANON     canon[2][MAX_CANON];

main() {
    screen(2,0,1,1);
    console(0,32,0);
    printf("¥33[>5h¥n");  /* カーソルを消す */
    while(1) {
  box(0,0,WIDTH-1,HIGHT-1,15,0xffff);
  initSinTable();
  initChar();
  while(1) {
      if(moveChar(0)) break;
      if(moveChar(1)) break;
      if(moveCanon()) break;
  }
    }
}

/*
    三角関数テーブルを作成する
*/
initSinTable() {
    int    i;

    for(i = 0; i < 256; i++) {
  sinT[i] = sin(2.0*PI/256.0*(float)i) * 100.0;
  cosT[i] = cos(2.0*PI/256.0*(float)i) * 100.0;
    }
}
/*
    キャラクタデータを初期化する
*/
initChar() {
    int    i,j;

    chdata[0].x = WIDTH/2+150; chdata[1].x = WIDTH/2-150;
    chdata[0].y     = chdata[1].y     = 200;
    chdata[0].velx  = chdata[0].vely  = 0;
    chdata[1].velx  = chdata[1].vely  = 0;
    chdata[0].rad   = chdata[1].rad   = 125;
    chdata[0].count = chdata[1].count = 0;

    for(i = 0; i < 2; i++) {
  for(j = 0; j < MAX_CANON; j++) {
  canon[i][j].exist = 0;
  }
    }
```

```
}

/*
    自分を動かす
*/
int moveChar(int i) {
    int         oldx,oldy,oldrad;
    int         acl = 0;    /* アクセル */
    int         brk = 0;    /* ブレーキ */
    char  kg2,kg3,kg4,kg7,kg8,kg9,kge;
    CHDATA      *p = &chdata[i];

    oldx = p->x; oldy = p->y; oldrad = p->rad;

    if(i == 0) {    /* プレイヤー1 */
  kg7 = BITSNS(7); kg8 = BITSNS(8);
  kg9 = BITSNS(9);

  if(kg8 & 0x10) acl = 1;           /* 8 */
  if(kg9 & 0x01) brk = 1;           /* 5 */
  if(kg8 & 0x80) p->rad++;          /* 4 */
  if(kg9 & 0x02) p->rad--;          /* 6 */
  if(p->triger) {
      if(!(kg7 & 0x20)) p->triger = 0;
      else ;
  } else {
      if(kg7 & 0x20) {    /* → */
          p->triger = 1;
          shoot(i , p);
      }
  }
    }

    if(i == 1) {    /* プレイヤー2 */
  kg2 = BITSNS(2); kg3 = BITSNS(3);
  kg4 = BITSNS(4); kge = BITSNS(0xe);

  if(kg2 & 0x04) acl = 1;           /* W */
  if(kg3 & 0x80) brk = 1;           /* S */
  if(kg3 & 0x40) p->rad++;          /* A */
  if(kg4 & 0x01) p->rad--;          /* D */

  if(p->triger) {
      if(!(kge & 0x02)) p->triger = 0;
      else ;
  } else {
      if(kge & 0x02) {    /* CTRL */
          p->triger = 1;
          shoot(i , p);
      }
  }
    }

    p->rad = (p->rad & 0xff);

    if(acl) {
  p->velx += (sinT[p->rad] / 10);
  p->vely += (cosT[p->rad] / 10);
    }
    p->vely++;    /* 重力 */

    if(p->velx < -1000) p->velx = -1000;
    if(p->velx >  1000) p->velx =  1000;
    if(p->vely < -1000) p->vely = -1000;
    if(p->vely >  1000) p->vely =  1000;

    if(brk) {
  p->x += p->velx / 400; p->y += p->vely / 400;
    } else {
  p->x += p->velx / 200; p->y += p->vely / 200;
    }

    if(p->x < 0) p->x = WIDTH;
    if(p->x > WIDTH) p->x = 0;
    if(p->y < 0) {
  bomb(i);
  return(1);
    }
    if(p->y > HIGHT-11) {
  if(abs(p->velx)+abs(p->vely) < 400 && abs(p->rad -125) < 5) {
  p->y = HIGHT-11;
  p->velx = 0; p->vely = 0;
  p->x = oldx; p->y = oldy;
  p->count = 10;  /* 弾の補給 */
  } else {
      bomb(i);
      return(1);
  }
    }

    putShip(oldx, oldy , oldrad, 0);
    if(i == 0)  putShip(p->x, p->y , p->rad, 5);
    else  putShip(p->x, p->y , p->rad, 7);
    return(0);
}

/*
    自機を描く
*/
putShip(int x, int y, int rad, int color) {
    line(x-sinT[rad]/10,y-cosT[rad]/10, x+sinT[rad]/6,
  y+cosT[rad]/6, color, 0xffff);
    line(x, y, x-sinT[(rad+32) & 0xff]/7,
  y-cosT[(rad+32) & 0xff]/7, color, 0xffff);
    line(x, y, x-sinT[(rad+224) &0xff]/7,
  y-cosT[(rad+224) &0xff]/7, color, 0xffff);
}

/*
    弾を発射
*/
shoot(int i, CHDATA *p) {
    int    j;

    if(p->count) p->count--;
    else    return;         /* 弾切れ */
    for(j = 0; j < MAX_CANON; j++) {
  if(canon[i][j].exist == 0 ) {
      canon[i][j].exist = 1;
      canon[i][j].x = p->x; canon[i][j].y = p->y;
      canon[i][j].rad = p->rad;
      break;
  }
    }
}

/*
    弾を動かす
*/
int moveCanon() {
    int         i,j;
    int         oldx,oldy;
    CANON *p;

    for(i = 0; i < 2; i++) {
  for(j = 0; j < MAX_CANON; j++) {
      p = &canon[i][j];
      if(p->exist == 0) {    /* 弾がない */
          timer(10);
          continue;
      }
      oldx = p->x;
      oldy = p->y;
      p->x += (CANON_SPEED * sinT[p->rad]);
      p->y += (CANON_SPEED * cosT[p->rad]);
      box(oldx, oldy, oldx+1, oldy+1, 0, 0xffff);
      if(p->x < 0 || p->x > WIDTH || p->y < 0 || p->y > HIGHT ) {
          p->exist = 0;    /* 弾は画面外へ */
      } else {
          box(p->x, p->y, p->x+1, p->y+1, 15 , 0xffff);
          if(abs(p->x - chdata[1-i].x)+abs(p->y - chdata[1-i].y)
                            < 10) {
              bomb(1-i);
              return(1);
          }
      }
  }
    }
    return(0);
}

timer(int time) {
    int    i,j;
    for(i = 0; i< time; i++ ){
  for(j = 0; j < 50; j++) ;
    }
}

/*
    爆発!!
*/
bomb(int num) {
    CHDATA *p;
    CANON dat[25];
    int    i;
    int    flag;
    char   ch[2];

    p = &chdata[num];
    for(i = 0; i < 25; i++) {
  dat[i].exist = 15; dat[i].rad = i * 10;
  dat[i].x = p->x;   dat[i].y = p->y;
    }
    putShip(p->x, p->y , p->rad, 0);
    do {
  flag = 0;
  for(i = 0; i< 25; i++) {
  box(dat[i].x, dat[i].y, dat[i].x+1, dat[i].y+1, 0, 0xffff);
      if(dat[i].exist) {
          dat[i].exist--;
          dat[i].x += cosT[dat[i].rad] * dat[i].exist / 20;
          dat[i].y += sinT[dat[i].rad] * dat[i].exist / 20;
          box(dat[i].x, dat[i].y, dat[i].x+1, dat[i].y+1,15, 0xffff);
          flag = 1;
      }
  }
    } while(flag);
    if(num == 0) symbol(150,150,"Player 1 Win",3,5,2,7,0);
    else   symbol(150,150,"Piayer 2 Win",3,5,2,5,0);
    do {
  locate(35 ,19);
  printf("Play Again (y/n)");
  b_inkey0(ch);
  ch[0] =toupper(ch[0]);
  if(ch[0] == 'N') {
      wipe();  printf("¥33*");      /* 画面クリア */
      printf("¥33[>5l¥n");          /* カーソルを表示 */
      exit(1);
  }
    } while( ch[0] != 'Y');
    wipe();  printf("¥33*");        /* 画面クリア */
}
```

任意の点を滑らかにつなぐ

高次方程式のグラフを描く

いくつもの点を指定して，それらを滑らかに結んでみよう
途中の点を得るためには複雑な高次方程式を解かねばならない
いったい，どのように処理するとよいのだろうか

Yokouchi Takeshi 横内 威至

タイトルはちょっとニュアンスが違うが，一応，高次方程式のグラフにはなっている。というのは，式を解いてグラフを描くというアプローチをとっていないのである。ではどういうことかというと，空間に適当な点を指定してやると，その点を通る曲線を描いてくれるという設定になっている。鼻水をたらしたガキは，「あ，ただのBスプラインじゃん」なんて，たかビる。でもこのサンプルは偉いことに疑似曲線ではなく完全曲線なのである。

サンプルリストの内容

汚らしいリストではあるが一応動くようである。X-BASICで，適当なファイル名でセーブしてやってもらいたい。RUNすると，まずXYZ軸に代わってRGB軸が表示されていることに気づく。そう，実は曲線はグラデーションでカラフルに描かれることになる。そこでまず通るべき点の数を指定する。2個から16個までだ。リスト中のx(16)，y(16)，z(16)さえ変えればいくらでも増やせるので自分でやってみよう。ただし単純計算で増やした分の2乗の時間がかかるので注意。次は全体で何ドット曲線上の点を計算するかをインプット。2なんか入れるといきなり始点と終点を結んで終わり。そしたら次は点を指定してやるのだがここはかなり低レベルである。ヘボいカーソルをカーソルキーで4方向，およびROLL UP，ROLL DOWNで前後に動かしてやり，リターンで決定。XYZ軸でなくRGB軸だから32段階であることがまた泣ける。改造でもするときはこれらは任意の実数でかまわない。全部入力したらあとは勝手に計算を始めてくれる。

画面右半分は，奥行きに応じて暗く描いている3Dぽいグラフィックが表示。左のケバいゾーンがグラデによる表示で，やや斜めから見ている。ただしクリッピングは一切していないから表示はエラく重なってしまう。またオーバーフローもチェックしてないのでヘタするとエラー。まあ，あまり気にすることはないが。

計算部分の説明

メインであり核心のこの領域は私の兄貴が考えたところであり，非常に感謝している。私ではこんな数学的センスは持ち合わせていないのが情けない。

高次方程式は，

$X=B_1X_1+B_2X_2+B_3X_3+\cdots\cdots B_nX_n$

$Y=B_1Y_1+B_2Y_2+B_3Y_3+\cdots\cdots B_nY_n$

$Z=B_1Z_1+B_2Z_2+B_3Z_3+\cdots\cdots B_nZ_n$

パラメータにtを使うと，

$t=-1$で，

$B_1=(-1)\hat{}a_1\times(t+1)t\ (t-1)\cdots\cdots(t-(n-2))/C_1$

$t=0$で，

$B_2=(-1)\hat{}a_2\times(t+1)t\ (t-1)\cdots\cdots(t-(n-2))/C_2$

以下$t=1, 2, \cdots\cdots n-2$まで同じようにB_nを作ることができる。ここで$t=-1$から始まっているのは実は無意味。0からでも1からでもよいのだ。こうなっているのはこれの前に作った疑似曲線の名残であり気にしないでほしい。以上が概念の内容だ。説明はあまりできないが，これが指定した各点を通ることは，それぞれのB_nにtを代入すればわかる。

と，書いてみたが私が考えたのではないから深いことはいえないのである。間違ったことをいってるかもしれないがたぶん平気だ。うーむ，やはり数学は難しく奥が深いらしい。まあとにかくこのプログラムは兄貴がメインをやってくれて，そこに私が泥を塗ったかたちのものであり，結構チョロい出来であろう。

一般的なBスプラインとの違いはわかるだろう。あれは疑似曲線にする代わりの利点として，スピードを高めているはずだ。

リスト

```
 10 screen 1,3,1,1
 20 float b,r,px,py,j,k,l,t,vx1,vx2,vx3,vy1,vy2,vy3,co1,co2
 30 dim float x(16),y(16),z(16),bb(16)
 40 str ke
 50 int c,c2,a,db,n,f,oldx,oldy,oldx2,oldy2
 60 vx1= 8:vy1=0:vx2=0:vy2= 8:vx3= 2/sqr(2):vy3= 4/sqr(2)
 70 sp_init():sp_clr(0,1):sp_disp(1):sp_on(0,1)
 80 setspr()
 90 n= 4:r=128
100 for a=0 to 31:pset(a*vx1,410,rgb(a,0,0)):next
110 for a=0 to 31:pset(0,410-vy2*a,rgb(0,a,0)):next
120 for a=0 to 31:pset(vx3*a,410-vy3*a,rgb(0,0,a)):next
130 pset(0,410,65535)
140 box(256,410,511,154,640)
150 paint(257,400,640)
160 setp()
170 sp_off(0,2)
180 cls:locate 0,0:print "n=";n,"point=";r
190 /*
200 /*
210 j=x(1):k=y(1):l=z(1)
220 oldx=vx1*j+vx2*k+vx3*l
230 oldy=410-vy1*j-vy2*k-vy3*l
240 oldx2=vx1*j+256:oldy2=410-vy2*k
250 printcurv()
260 end
270 /*
280 /*--------MAIN--------
290 func printcurv()
300 for i=0 to r
310 t=i*(n-1)/r-1
320 getcurv()
330 conv3d()
340 line(oldx,oldy,px,410-py,c)
350 pset(px,410-vy3*l,990)
360 pset(vx3*l,410-py,30750)
370 line(oldx2,oldy2,256+vx1*j,410-vy2*k,c2)
380 oldx=px:oldy=410-py
```

そしてどこを“疑似”とするかというと，指定点を結ぶ直線である。このプログラムはそれとはまったく異なるものだ。完全曲線ではあるが，ひとつの式で表される曲線であり，点が多いほど曲線は予想からはずれることになる。制御することは不可能に近い。さすがに4点ぐらいなら単純な方程式になり，予想は容易である。パラメータの変域を変えればいろいろな曲線にはなるはずだが，点を結ぶ直線に近いものを得ることはほぼない。まあ，本来の目的とは違うことから生まれたプログラムであり，これはこれでよいとしている。

裏の事情

私はグラフを描かせたいがためにこれを作ったのではない。元は，グラデーションデータを勝手にはじき出すものを作ろうと思っていた。2色間のグラデーション，Z'sSTAFFのやつとかでは不満が多いのだ。ありがちな例をあげると，青っぽいメタリックなオブジェクトを描くとき，白－青より白－水色－青なんかを多用する。そう，つまり途中の段階をつけてグラデーションしてくれるやつがほしいのだ。そして変化率も変えたいのだ。そのまま描いて球に見えるようにだとか楕円のようにだとか，だ。このルーチンを使えば結構微妙な調整も可能だと読んでいる。

またアニメーションなどのモーションをはじきだすことなんかにも応用できそうだ。とりあえずグラデーションのやつでも作ってみようと思う次第である。

ところでX-BASICを使ったのはほぼ初めてであった。だからプログラムはかなりヘボいのである。ああ，私の恥ずかしいところを見られてしまう……。

＊　　＊　　＊

編集部の人はストII強すぎる。順位ではケツになってしまった。しかもいくらやってもうまくならん。まいったな。ピットファイターなら誰にも負けないのだがあんなものに手を染める人間はほかにはいないのだろうか。格闘とはもっと泥臭く，容赦ないものだ。そんな香りのあるゲームはこれぐらいなものなのに。3人同時プレイの熱さを誰か知らないか？　敵同士も殴りあう，あのクレイジーなムードを……。残虐行為手当！　誰がこんな単語を創造できるだろうか。アタリの日本語は鋭く心に響く。

```
390 oldx2=vx1*j+256:oldy2=410-vy2*k
400 next
410 endfunc()
420 /*--------2D to 3D--------
430 func conv3d()
440 px=vx1*j+vx2*k+vx3*l
450 py=vy1*j+vy2*k+vy3*l
460 s=((j<0)+1)*j+((j-31)>0)*(j-31)
470 t=((k<0)+1)*k+((k-31)>0)*(k-31)
480 u=((l<0)+1)*l+((l-31)>0)*(l-31)
490 c=rgb(s,t,u):c2=rgb(31-u,31-u,31-u)
500 endfunc()
510 /*--------!!--------
520 func getcurv()
530 for a=1 to n
540 bb(a)=1
550 f=1-(n-n/2*2)
560 for db=1 to n
570 if db<>a then {
580  co1=a:co2=db
590 bb(a)=bb(a)*(t+2-co2)/abs(co1-co2)
600  }
610 next
620 co1=f+1+a
630 co2=1-(co1-int(co1/2)*2)*2
640 bb(a)=bb(a)*co2
650 next
660 j=0:k=0:l=0
670 for a=1 to n
680 j=j+x(a)*bb(a)
690 k=k+y(a)*bb(a)
700 l=l+z(a)*bb(a)
710 next
720 endfunc()
730 /*--------input--------
740 func setp()
750 input "input numbers(2-16)",n
760 n=((n<2)+1)*(n-2)+2+((n-16)>0)*(n-16)
770 input "how many dots?",r
780 cls
790 for a=1 to n
800 locate 0,0:print a;"/";n
810 c2=0
820 while c2=0
830  sp_move(2,j*vx1+l*vx3,410-k*vy2-l*vy3,0)
840  sp_move(0,j*vx1+l*vx3,410-l*vy3,1)
850  sp_move(1,l*vx3,410-k*vy2-l*vy3,1)
860 locate 0,2:print "x=";j
870 locate 0,3:print "y=";k
880 locate 0,4:print "z=";l
890  ke=inkey$:c=asc(ke)
900  switch c
910  case 13:x(a)=j:y(a)=k:z(a)=l:c2=1:break
920  case 28:j=j+1:break
930  case 29:j=j-1:break
940  case 30:k=k+1:break
950  case 31:k=k-1:break
960  case 14:l=l-1:break
970  case 15:l=l+1:break
980  endswitch
990  j=((j<0)+1)*j+((j-31)>0)*(j-31)
1000  k=((k<0)+1)*k+((k-31)>0)*(k-31)
1010  l=((l<0)+1)*l+((l-31)>0)*(l-31)
1020 endwhile
1030 pset(j*vx1+l*vx3,410-k*vy2-l*vy3,rgb(24,31-l,31-l))
1040 next
1050 endfunc
1060 /*---------SPRITE-----------
1070 func setspr()
1080 sp_color(2,65535,1):sp_color(1,49854,1)
1090 dim char sp0(255)={
1100       1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1110       1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1120       1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1130       1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
1140       1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
1150       1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,
1160       1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,
1170       1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,
1180       1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,
1190       1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,
1200       1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,
1210       0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,
1220       0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,
1230       0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,
1240       0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,
1250       0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0
1260 }
1270 dim char sp1(63)={2}
1280 sp_def(0,sp0,1):sp_def(4,sp1,0)
1290 endfunc()
```

電源システムの管理

パワーダウンマネージャ

使い込めば使い込むほど，システムに対する不満が出てくる
その不満点を解消するために，どのような解決方法があるのだろうか
今回，自分好みのシステムを構築するために，瀧氏がとったアプローチとは？

Taki Yasushi 瀧 康史

システムは広がりをもつものであってほしい。このままの環境でいてほしくない。徐々に広がりをもって大きく，そして葉を広げてほしい。こんな願いは，たとえひとりのパーソナルユーザーだとしてももっているはず。自分の持っている環境を少しでもよいものにしてみたいと。

なにを作るか

今回目をつけたのは，X68000の電源処理です。たとえば，電源を切るときは，RAMディスクからあれとこれをハードディスクに収め，この処理を終えてから電源を落としたい，そんな作業が必要なときがあるかもしれません。

こういった願いをもった人は私のほかにもいるわけで，電源OFFでファイルをコピーするプログラムや，電源をソフトウェア的に落とすプログラムもいくつか発表されています。

しかしながら，電源OFFでファイルをコピーするプログラムは，RAMディスクのリストア程度にしか使えないでしょう。私が知っているかぎり，ソフトウェア的に電源を落とすプログラムは，常駐する（厳密には常駐はしない）とき，フロント電源スイッチを手動で切る（しかし電源は落ちない）という作業が必要になります。この処理は，たいていAUTOEXEC.BATを実行したときに行うので，起動してから2～3分もかかるくらい肥大したシステムになると，システムが起動する途中にこの処理が入り，非常にうっとうしくなります。

そこで，これらのうっとうしい作業から解放するため，手動で電源スイッチをOFFにする処理を起動直後，すなわちSRAM上で行うことにします。電源OFFのときは，POFF.BATなどを作り，やりたい処理をすべてバッチ中にやらせ，最後にリセットを実行することにより電源をOFFにするようにします。これでずいぶんと作業を円滑に進めることができるでしょう。

X68000ではリセットがかかると，外部のEXP-ON信号が入っているか，タイマ起動中か，フロント電源スイッチが入っているかのいずれかが満たされるとブートします。プログラム中でフロント電源スイッチを切ってあるので，リセットがかかると電源が落ちるというわけです。当然ながら自作プログラム中でリセットを行っても電源を切ることができます。

なお，常駐といってもベクタをフックし，もう一度元の状態に戻すので，厳密には常駐しているわけではありません。したがって，このプログラムにより，動かないソフト出てきたりはしないでしょう。

入力方法

用意したプログラムは3つです。まず，リスト1がX68000にリセットするための「TRAP10.S」，リスト2がSX-WINDOW ver.2.01で終了時にリセットをかけるためのパッチ当てプログラム，リスト3がSRAM上に常駐するSRAM.Sです。

入力方法は，リスト1のTRAP10.Sは普通にアセンブラと，リンカを通してやればOKです。CVでR形式にしておきましょう。

リスト2は見てのとおりBASICで書かれたプログラムで，カレントディレクトリにSXWIN.Xを用意してRUNするだけです。

これによって，SX-WINDOWが終了したときにタイトルが赤くなって無限ループに入らずリセットがかかり，SRAM.Xを組み込んでおけば終了と同時に電源が落ちます。当然，SX-WINDOWをコマンドから呼び出したときは，電源が落ちませんし，SRAM.Xが組み込まれずに実行されるとリセットがかかります。

リスト3のアセンブル方法は，リスト3のコメントにも書いてあるとおり，

AS −W SRAM.S

以上のようにすればOKです。アセンブラにAS.X ver.2.0かHAS.Xを使用してください。AS.X ver.1.0でも一応動作しますが，余計なコードを吐き出しまいますので注意が必要です。

リンカは，LK.X ver.1.0でもver.2.0でもかまいません。ただし，HLKはBオプショ

SRAMディスクあれこれ

ここで，SRAMディスクにインストールする際に発生する問題を解決するためには，どうしたらいいか？ その方法を紹介しましょう。

まず，プログラムが連続したトラックに転送されない場合は，SRAMディスクの容量以上のファイル（13Kバイト以上）をいったんSRAMにコピーさせます。もちろん容量以上のものをコピーするわけですから，エラーが発生します。しかし，とりあえず無視。で，そのあと目的のプログラムをコピーしてやると，連続したトラックにプログラムが転送されるはずです。

もうひとつ，SRAMの内容（$ED0100～$ED3FFF）を0クリアするプログラムを用意しました。SRAMディスクをフォーマットしても，管理領域のみ初期化されるので，本当の意味でSRAMを初期化したい場合に使用してください。

なお，このプログラムはSRAMディスクの管理領域までクリアするため，プログラム実行後，SRAMディスクを登録していてもシステムに認識されなくなるので注意しましょう。SRAMディスクを使いたい場合は，再びフォーマットをかける必要があります。インストールがうまくいかない人は，このプログラムでSRAMをきれいにしてからインストール作業をやり直してみてください。

```
*SRAMの内容をクリアする
*       ($ED0100～$ED3FFFF)
        .include  doscall.mac
        .include  iocscall.mac
        .text
entry:
        lea.l     0,a1
        IOCS      _B_SUPER
        move.b    #$31,$e8e00d
        lea.l     $ed0100,a2
        move.w    #$1f7f,d7
loop:
        clr.w     (a2)+
        dbf       d7,loop
        move.b    #0,$e8e00d
        move.l    d0,a1
        IOCS      _B_SUPER
        DOS       _EXIT
        .end
```

ンがないようなので（あるかもしれませんがわかりませんでした）使えません。リンク方法は，

LK －B ED0C40 SRAM.O

これでSRAM.Xができます。

また，リスト３の34行目にあるラベル，dbgを１に書き換えることでコマンドモードから実行できるテストモデルにもなります。あと，114～118行の注釈を無効にすることで，起動時にAD PCMデータを鳴らすことができます。AD PCMデータは，247行からの注釈行を参考にしてください。

インストール

基本的には先月のSAVESC.SYSと同じ方法でインストールできますが，先月号を買ってない人のために一応説明しておきましょう。

まず，SRAMDISK.SYSが定義されたシステムを立ち上げます。メモリスイッチのSRAMの利用をRAMDISKにして，CONFIG.SYSに，

DEVICE＝SRAMDISK.SYS

を，加えてリセットをすればOKです。システムが立ち上がったら，SRAMDISKのドライブを調べます。SRAMDISKもDRIVE命令ではRAMDISKと表示されますが，なにも入ってなければSRAMDISKは全容量13Kバイトですのですぐにわかるでしょう（まさか，RAMDISKを13Kバイトしかとってない人っていないよね？）。

SRAMDISKにプログラムが入っている人は，まずフォーマットをしてください。当然ですが，いままでのSRAMの中身は消えてなくなってしまうので，消えて困る場合はバックアップをとっておきましょう。

フォーマットが終わったら，入力したプログラムをSRAMDISKにコピーします。しかし，ここで気をつけなくてはならないのは，プログラムがちゃんとSRAMの先頭，かつ連続したトラックに転送されたかどうかです。これをチェックするには，

CHKDSK －A？:

（？はSRAMDISKのドライブ番号。各自変更すること）とコマンドから打ち込みます。そのとき，

A----- SRAM.X $00000003～$00000003

と表示されていなくてはなりません。最初の$03はSRAMDISKの最初のトラックに入っているということ。このプログラムは１クラスタ以内に入っているのでかまわないのですが，もし長いプログラムの場合は，これらが連続されているかどうかチェックしてください。

滅多に起きませんが，以下のように出たらインストールは失敗です。

A----- SRAM.X $00000004～$00000004

失敗した場合は，何度かフォーマット作業をやり直してください。次に，SRAMブートにします。コマンドラインから，

SWITCH B＝RAM1

とやっても，SWITCHを起動して変えてもOKです。心配なら，デバッガなどでED0010H番地が00ED0C40Hに，ED0018H番地がB000Hになっているか確認してみましょう。これでインストールは終了です。

使用方法

リセットすると，ラストメモリチェックを行います。このチェックはいわゆる本当の意味でのメモリチェックではありません（厳密なメモリチェックはそのアドレスでプログラムを実行したりしなくてはなりません）。

このラストメモリチェックというのは，RAMの最終番地を自動的に判断し，SRAMDISKに書き込みます。要するに，RAMを増設してもSWITCHで書き換える必要がなく，自動的に最大メモリに設定されるのです。

拡張スロットにメモリが差さってる人は，試しに抜いたり差したりしてみるとわかります。物理的にあるところまでしかメモリチェックはしません。そして，同時にSRAMのメモリスイッチも変更されているのが確認できるでしょう。

メモリチェックが終了すると，最終メモリ番地，フリーエリアが画面に表示されます。

これらの表示が出たら，フロントの電源スイッチを切って（アンフック）ください。いつもどおり，電源ランプが点滅しますが，無視すること。しばらくすると，“BOOT!”という文字が表示され，通常起動します。あとはリセットがかかるまで電源は落ちません。

なお，このシステムはリセットと同時に電源がOFFされるので，リセットを頻繁にする場合には不向きです。

ハードディスクを使用している場合は，一度立ち上げたら仕事が終わるまでリセットせずにすみますが，ゲームなどをやる場合は，頻繁にリセットをかけます。また，ハードディスクを使っていても暴走する可能性のあるソフトを開発している場合は，リセットを頻繁にかけるでしょう。そのときは，CTRLキーを押したままリセットをかけることで電源スイッチを切らずに再起動します。

では，自分の気に入ったシステムを作るためにこのプログラムを役立ててください。

リスト1

```
*  X68000にリセットをかける
        .text
        trap    #10
        .end
```

リスト2

```
10 int a:a=fopen("sxwin.x","rw")
20   fseek(a,&H153,0)
30   fputc(&H4A,a)
40 fclose(a)
```

リスト3

```
 1: *     Power Down Manager Ver 2.01
 2: *        アセンブル       AS -W SRAM.S
 3: *        リンク           LK -B ED0C40 SRAM.O
 4: *                         で可能。HLKは使用できません。
 5: *
 6:          .include         iocscall.mac
 7:          .include         doscall.mac
 8:
 9:          .text
10:
11: dbg:     equ      0                        * 1のときテストモデルになる
12:
13:          bra      mainprog                 * SRAMに格納するには必ず必要
14:          nop
15:
16: mainprog:
17:
18:          movem.l d0-d7/a0-a6,-(SP)
19:
20:          if dbg=1                          * テストモデルは、スーパバイザモード
21:          clr.l    -(SP)                    * にしてコマンド起動を許可。
22:          DOS      _SUPER
23:          addq.l   #4,sp
24:          move.l   d0,SSPBUF
25:          bra      vectrap
26:          endif
27:
28:          move.b   #$31,$e8e00d             * SRAM書き込み許可
29:
30:          lea      memchk,a0
31:          tst.b    (a0)                     * メモリチェック後の起動なら
32:          beq      vectrap
33:
34:          move.w   #$0002,d1                * バスエラーフック
35:          lea      ERR,a1
36:          IOCS     _B_INTVCS
```

```
        move.w  #$0,d1                  * x=0
        move.w  #$0,d2                  * y=0
        IOCS    _B_LOCATE
        lea     memchking(PC),A1        * Memory Checking...
        IOCS    _B_PRINT
        move.b  #$30,d1
        IOCS    _B_PUTC                 * 0
        lea     memchking2(PC),A1       * M Bytes
        IOCS    _B_PRINT
        move.l  #$00010000,a6           * メモリーチェック
        clr.l   d5
        move.l  #$55555555,d4           * ダミーデータ
AA:     move.l  (a6),d7                 * もとのデータを保存
        move.l  d4,(a6)                 * データを書く（$5eは意味なし）
        cmp.l   (a6),d4                 * 同じか?
        bne     ERR                     * メモリーエラーにとぶ。
        move.l  d7,(a6)                 * 保存されていたデータを戻す
        move.l  a6,d6                   * d6は現在のアドレス
        addq.l  #$4,a6                  * 画面表示のためのチェック
        and.l   #$000fffff,d6
        cmp.l   #$000ffffc,d6
        bne     AA
                                        * 表示ルーチン
        move.w  #$0,d1                  * x=0
        move.w  #$0,d2                  * y=0
        IOCS    _B_LOCATE
        lea     memchking(PC),A1        * Memory Checking...
        IOCS    _B_PRINT
        add.l   #$1,d5
        move.l  d5,d0                   * value
        lea     strings,a0              * pointer
        move.l  #10,d2
        bsr     _itoa
        lea     strings(PC),A1          * n
        IOCS    _B_PRINT
        lea     memchking2(PC),A1       * M Bytes
        IOCS    _B_PRINT
        bra     AA
ERR:
        move.l  a6,$0ED0008             * エラーの出たアドレスを
                                        * 最終アドレスにする
        lea     memchk,a0
        move.b  #$00,(a0)
        trap    #10                     * リセット
vectrap:
        lea     memchk,a0
        move.b  #$ff,(a0)
        lea     Title(PC),A1            * メッセージの表示
        IOCS    _B_PRINT
        lea     FreeArea(pc),A1         * Free Area.......
        IOCS    _B_PRINT
        move.l  $0ED0008,d0
        move.l  #10,d2
        lea     strings,a0
        bsr     _itoa
        lea     strings(PC),A1          * n
        IOCS    _B_PRINT
        lea     Bytes(pc),A1            * Bytes.
        IOCS    _B_PRINT
        lea     LastAddres(pc),A1       * Last Address....
        IOCS    _B_PRINT
        move.l  $0ED0008,d0
        subq.l  #$1,d0
        move.l  #16,d2
        lea     strings,a0
        bsr     _itoa
        lea     strings(PC),A1          * m
        IOCS    _B_PRINT
        lea     Hex(pc),A1              * Hex.
        IOCS    _B_PRINT
*       moveq   #$60,d0                 * ADPCM再生ルーチン
*       move.w  #4*256+3,d1             * 4:15.6kHz 3:左右同時出力
*       move.l  #pcmend-pcmdata,d2      * AS Ver 1ではここで余分なコードを吐き出す
*       lea     pcmdata(PC),a1          * データ領域設定
*       trap    #15
        move.w  #$002a,d1               * 電源フックルーチン
        lea     return,a1
        IOCS    _B_INTVCS
        move.l  d0,vecset
BB:     move.w  #$0e,d1
        IOCS    _BITSNS
        and.w   #$02,d0                 * CTRLが押されていたら、
        bne     CC                      * 正常終了へ、
        lea     veccheck0,a1            * VECCHECKが
        tst.b   (a1)                    * 0ならば
        beq     CC                      * 正常終了へ、
        bra     BB                      * そうでなければ、ループ
CC:                                     * 正常終了！
        move.b  #$31,$e8e00d            * SRAM書き込み許可
        lea     veccheck0,a1            * VECCHECK0 の値をもとに戻す。
        move.b  #$ff,(a1)               * これが無いと困るでしょ。
        move.w  #$002a,d1               * trap #10 フック
        move.l  vecset,a1
        IOCS    _B_INTVCS
        lea     Boot(pc),A1             * Boot!
        IOCS    _B_PRINT
        move.b  #$00,$e8e00d            * SRAM書き込み禁止
        if      dbg=1
        move.l  SSPBUF,-(SP)
        DOS     _SUPER
        addq.l  #4,SP
        DOS     _EXIT
        endif
        movem.l (SP)+,d0-d7/a0-a6       * スタックからレジスタを戻す
        rts                             * これで、STDブートする
return:
        move.b  #$31,$e8e00d            * SRAM書き込み許可
        lea     veccheck0,a1
        clr.b   (a1)
        rte
*       _ltoa
*       _itoa
*       _wtoa
*
*       d0.l にvalue 変換すべき整数値
*       a0.l に文字列の格納領域を示すポインタ
*       d2.l に基数
_ltoa:
_itoa:
        bra     _toa0e
_wtoa:
        ext.l   d0
_toa0e:
        clr.b   (a0)
        movea.l a0,a1
        cmpi.l  #$00000002,d2
        blt     _toa62
        cmpi.l  #$00000024,d2
        bgt     _toa62
        lea.l   -$0080(SP),SP
        movea.l SP,a2
        clr.b   (a2)+
        cmp.b   #$0a,d2
        bne     _toa42
        tst.l   d0
        bge     _toa42
        move.b  #$2d,(a0)+
        neg.l   d0
_toa42:bsr      _toa66
        addi.b  #$30,d1
        cmp.b   #$3a,d1
        bcs     _toa50
        addq.b  #7,d1
_toa50:move.b   d1,(a2)+
        tst.l   d0
        bne     _toa42
        move.l  a1,d0
_toa58:move.b   -(a2),(a0)+
        bne     _toa58
        lea.l   $0080(SP),SP
        rts
_toa62:
        move.l  a1,d0
        rts
_toa66:
        move.w  d0,d1
        clr.w   d0
        swap.w  d0
        divu.w  d2,d0
        swap.w  d0
        swap.w  d1
        move.w  d0,d1
        swap.w  d1
        divu.w  d2,d1
        move.w  d1,d0
        swap.w  d1
        rts
        .even
        .data
veccheck0:
        .dc.b   $ff
memchk:
        .dc.b   $ff
memchking:
        .dc.b   ' Memory Checking...',0
memchking2:
        .dc.b   ' M Bytes',$0d,$0a,0
Title:
        .dc.b   ' Process Manager Ver2.06 Copyright 1992 by Kohju'
        .dc.b   $0d,$0a,$0d,$0a,0
FreeArea:
        .dc.b   ' Free Area.......',0
Bytes:
        .dc.b   ' Bytes.',$0d,$0a,0
LastAddres:
        .dc.b   ' Last Address....$',0
Hex:
        .dc.b   '(Hex.)',$0d,$0a,0
Boot:
        .dc.b   ' Boot!',0
        .even
pcmdata:
*       データ部です。ADPCMデータを、いろいろいれて下さい。
*       .DC.B   $??,$??
*       ようにかけばOKです。
*       尚、私は、DISで、
*       DIS [ADPCM FILE NAME] [なんでもいい] /z0 0
*       で、適当なソースを作成し、エディタで挿入します。
*       あとは余分なところを切って、出来上りです。
pcmend: .dc.b   0
        .bss
        .even
vecset:
        .ds.b   4
VecFook:
        .ds.w   1
strings.
        .ds.b   32
if dbg=1
SSPBUF:
        .ds.b   4
endif
        .end
```

電卓インベーダゲームを再現する

あなたと私の電卓物語

Nishikawa Zenji 西川 善司

いまとなっては，懐かしい電卓インベーダゲームを紹介しよう
ゲームは，迫りくる数字インベーダを撃墜するだけのシンプルさ
ルールが簡単だからこそ，とっつきやすく熱中できるゲームだ

私がまだパソコンを知らない時代，一世を風靡したゲーム機があった。ファミコンもまだ世に出ていない頃の話だ。

当時の私は日本を遠く離れ，異国の地にいた。日本の新しい玩具やマンガ本は，異国に住む日本人の小僧たちにとっては，このうえない興味羨望の的となった。猫なで声で手招きする美女と，コロコロコミックを持った全裸の金○氏がいたとしたら，当時の私なら迷わず，金○氏のところへ走っただろう(10年後後悔するかもしれないが)。

話を戻して，当時，ひとりの友人Iが，なにやら電子計算機をかちゃかちゃいじりながら白熱していた。小学生のくせに電卓を持ち歩いているとはマセたガキだ，と自分も小学生であることをすっかり忘れ，ナマイキなことを考えながら近くに寄っていき，I君の背後から電卓をのぞき込んだ。効果音と，I君の熱中の形相から一瞬にしてゲームであることは予想がついたが，まったくルールがわからなかった。さっそく，「Iちゃん，やり方教えてよ。やらせてよ」とねだった。彼は素直で，すぐに説明をしてくれたうえに1日貸してくれるとまでいった。

電卓インベーダゲームのルールはいたって単純だった。右から出現し迫りくる数字を左ボタン(セレクトボタン)でセレクトし，消したい数字に合わせ右ボタン(ファイアボタン)で撃墜するだけ。位置が右の数字ほど得点は高く，消去した数値を加算していき，これが10の倍数になると“Π”のような形のUFOが現れる。これを消去すれば高得点。数字列がいちばん左まで接近してしまうと1ミス。3ミスでゲームオーバー。弾の持ち数は30，敵は16。つまり16個の数字を消去すれば1面クリア。また，ムダ撃ちをしすぎて弾を使い切ってしまってもゲームオーバーとなる。1面クリアすると残機数や弾数も元に戻される。面が進むごとに数字の接近スピードは増していく。手早い指さばきと暗算の速さが高得点への道へとつながる。なかなかのアイデアゲームだ。

私には当時，面白いものはなんでも風呂に入りながら楽しむという，貧乏人のくせになかなか高貴な趣味があった。ご多分にもれずその日も電卓インベーダーを風呂に入りながら「ピロロロ」などと楽しんでいたのだが，手が滑り浴槽の中に落としてしまうというハプニングを引き起こしてしまった。不本意に風呂に入ってキレイにされた電卓君は，文字は出てもボタン操作に反応しなくなっていた。分解してタオルで拭いてみたらなんとか動くようになり「ふぅ」っと胸をなでおろし，天にまします神様に感謝などをしてしまった。

そして，現在，ついに買った念願のコードレス電話の子機を風呂場に持ち込むことが私の毎日の楽しみのひとつとなっている。「反省」という熟語の意味をよく解さない男Z.Nの寂しい物語だ。

プログラム

ネタに詰まった私は，結局電卓インベーダゲームをお届けすることになった。ゲームプログラムはX-BASICで書かれており効果音を鳴らすためにOPMDRV.X，またはZMUSIC.Xの組み込みを必要とする。

X-BASICはなかなか美しいプログラムを書ける利点を備えている反面，初めからプリセットされているコマンドの数が異様に少ない欠点もある。今回，どうしても外字定義とキーリピートなしのキー入力が必要であったため（ウソ度50%)，これらの命令を外部関数として別に作成した（リスト1)。そして，ゲームプログラムがリスト2となっている。

プログラムの入力

リスト1はX-BASICの外部関数である。これはアセンブラ記述言語で書かれている。よって実行にはアセンブラAS.XやリンカLK.X，またはそれらの同等品が必要になる。

リスト1をED.Xなどのエディタでファイル名“BIT.S”として入力し，これを，

　A>AS BIT

としてアセンブル，

　A>LK BIT

としてリンクし，

　A>REN BIT.X BIT.FNC

で外部関数としてリネームする。これを自分のディスクのBASICディレクトリにコピーし，エディタで“BASIC.CNF”を読み込み，

　FUNC＝BIT

の1行を追加しエディタを終了。このあとX-BASICを起動して，

　BITSNS(0)

を実行してみよう。

　Ok

が表示されずになんらかのエラーが出た場合はもう一度間違いがないか，リスト1をチェックしてみよう。

リスト1がうまくいけばあとは簡単。X-BASIC上からリスト2を入力して，

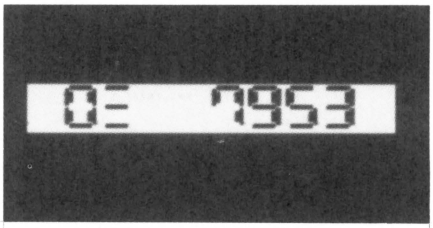
　RUN

で実行するだけ。

ゲームの遊び方

ルールは冒頭に書いてあるとおり。キー操作はカーソルキーの“←”“→”の2つだけ。“←”が数字のセレクトキー，“→”がショットボタン。これだけだ。ゲームオーバーになったら“0”キーで再プレイができる。

プログラムをRUNすると，ゲーム画面の大きさを聞いてくるので，大きくしたい場合は“Y”キー，小さくてよければそれ以



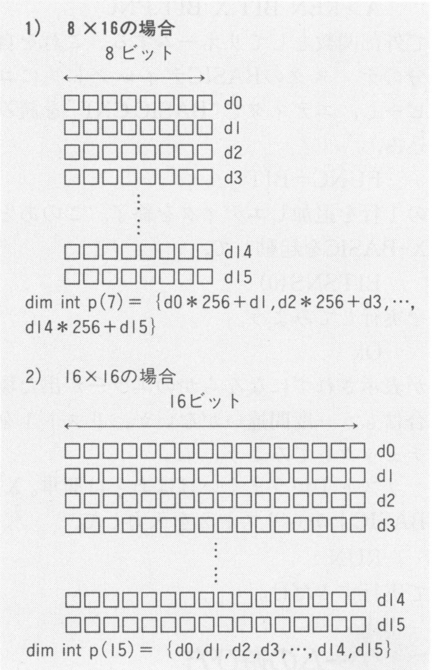
外のキーを押そう。ハイスコア，ノルマの表示後ゲームは開始される。

やりだすと結構ハマる。全9面で9面をクリアすると1面に戻るループ構成。みんなは何点いくかな。

プログラムの説明

最近はアセンブラ上でプログラムを書くことが多かったので，ジャンプ(GOTO文)が実質上使えないX-BASICでのプログラミングは非常にメンドくさかった。リストはご覧のとおり「美しい」GOTO文なしのものに仕上がっている……といいたいが，私はWHILE文やREPEAT文だらけのプログラムをそれほど美しいと感じないヒネクレ者なので，なんかあと味がよくない。

図1 パターンの構造



プログラムをざーと見ていくと，半分が数字パターンの定義，効果音の定義に取られているのがわかるだろう。

メインループで敵の出現および敵の撃墜処理を行い，面クリア，ミス，ゲームオーバーなどの処理はメインループを抜けて行っている。

プログラムの詳しい説明は省略して，使われている変数について少し解説しておこう。max_lは敵の数字軍団を何匹画面に出すかを決定している。mspはゲームのスピード。面が進むにつれてmspの値が減っていき，ゲーム全体のスピードが速くなっていく。shは残機数，bは弾数，enは1面当たりに出現する敵数字の数，yはゲーム画面をディスプレイのどの行に表示するかを決定するもの……と，こんな感じ。ゲームが難しいと思ったらmspやmax_lの値を大きくするといいだろう。

BIT.FNCの使い方

今回作成した外部関数の使い方を簡単に説明しておこう。X-BASICでなにかを作るときに役立つかもしれない。

●bitsns(n)

[引数]

・n：キーコードグループ($00～$0F)

キーバッファに無関係のリアルタイムキー入力関数。さらに同時に複数のキーが押されても判別が可能。押されたキーコードグループを引数とし，これに含まれるキーの押下状態をビットで返す(BIT＝1でキーが押されている，BIT＝0で押されていない)。キーコードグループと対応キーの関係は，プログラマーズマニュアル259ページの表を参考にしてもらいたい。たとえばカーソルキーの“↑”を調べたい場合，

a＝bitsns(7)

とし，

a and &b00001000

が1のとき押されている，0のとき押されていない，となる。さらに，

a and &b00000010

が1のときはカーソルキー“↓”が押されていることになるわけである。

●set_gaiji(n,p)

[引数]

・n：漢字コード

・p：定義パターン，int型1次元配列

pのパターンで漢字コードnの外字を定義する。X1シリーズでいうDEFCHR$命令に相当する。定義できる外字は16×16ドットまたは8×16ドットのパターンのみ。データ形式は図1に示すとおり。16×16ドットの場合の例をリスト3に示す。

終わりに

音楽以外の特集記事に参加するのって久しぶりのような気がする。なんかすごく新鮮だ。また，時間の許すかぎり参加していきたい。それでは，みなさん応援よろしく。ごきげんよう，さよなら。

リスト1

```
*                                               *
*         リアルタイムキーチェックと            *
*         外字定義の外部関数                    *
*                                               *

          .include      doscall.mac
          .include      iocscall.mac
          .include      fdef.h

information_table:
          dc.l    init
          dc.l    run
          dc.l    end
          dc.l    system
          dc.l    break
          dc.l    ctrl_d
          dc.l    yobi
          dc.l    yobi
          dc.l    token_table
          dc.l    parameter
          dc.l    exec_address
          dcb.b   20,0

token_table:
          dc.b    'bitsns',0                    *1
          dc.b    'set_gaiji',0                 *2
          dc.b    0                             *end code
          .even
parameter:
          dc.l    bitsns_p                      *1
          dc.l    set_gaiji_p                   *2

bitsns_p:
          dc.w    char_val                      *1
          dc.w    int_ret
set_gaiji_p:
          dc.w    int_val,ary1_i                *2
          dc.w    int_ret

init:
run:
end:
system:
break:
ctrl_d:
yobi:
          rts

exec_address:
          dc.l    bitsns                        *1
          dc.l    set_gaiji                     *2

bitsns:
          move.l  12(sp),d1
          cmpi.w  #$0f,d1
          bhi     err1
          IOCS    _BITSNS
          bra     ok_ret

set_gaiji:
          move.l  12(sp),d1
          ori.l   #$0008_0000,d1
          move.l  22(sp),a2
          addq.w  #4,a2
          tst.w   (a2)+
          bne     err2
          cmpi.w  #4,(a2)+
          bne     err2
          move.w  (a2)+,d0
          cmpi.w  #15,d0
          bhi     err2
          lea     gj_buf(pc),a1
sg_lp:
          move.l  (a2)+,d2
          move.w  d2,(a1)+
          dbra    d0,sg_lp
          lea     gj_buf(pc),a1
          IOCS    _DEFCHR

ok:                                     *戻り値無しでリータン
```

```
        moveq.l #0,d0
ok_ret:                         *戻り値有りてリータン
        lea     ret_buf(pc),a0  *戻り値ポインタ設定
        move.l  d0,6(a0)        *戻り値書き込み(low long word)
        moveq.l #0,d0
        rts

err1:
        moveq.l #1,d0
        bra     error
err2:
        moveq.l #2,d0

max_err:        equ     (emte-err_mes_tbl)/2
error:                          *エラー発生時の処理
        lea     ret_buf(pc),a0  *戻り値ポインタ
        move.l  d0,-(sp)
        cmpi.l  #max_err-1,d0
        bls     er0
        moveq.l #3,d0           *unknown(=3)
er0:
        add.w   d0,d0
        move.w  err_mes_tbl(pc,d0.w),d0 *エラーメッセージポインタ
        lea     err_mes_tbl(pc,d0.w),a1
        move.l  #-1,6(a0)
        move.l  (sp)+,d0
        rts

err_mes_tbl:
        dc.w    0
        dc.w    err_mes1-err_mes_tbl    *1
        dc.w    err_mes2-err_mes_tbl    *2
        dc.w    err_mes3-err_mes_tbl    *3
emte:

err_mes1:       dc.b    'bitsns命令の使用法に誤りがあります',0
err_mes2:       dc.b    'set_gaiji命令の使用法に誤りがあります',0
err_mes3:       dc.b    '正体不明のエラーです',0

        .even
ret_buf:        dc.w    0       *+0
ret_buf_h:      dc.l    0       *+2 hi
ret_buf_l:      dc.l    0       *+6 low
        .bss
gj_buf:         ds.w    32
```

リスト2

```
  10 /******************************/
  20 /*                            */
  30 /*    あなたと私の電卓物語    */
  40 /*                            */
  50 /*     Programmed  By Z.N     */
  60 /*                            */
  70 /******************************/
  80 color 3:width 64:console 0,31,0
  90 key 9,"color 3:width 96"+chr$(13)
 100 key 19,"save"+chr$(34)+"den"
 110 /*** いわゆる初期設定 ***/
 120 str e[256],m,kh[256],dmy,ini[256],s
 130 dim str  o(13)
 140 dim char v(4,10)
 150 dim int  gj(7)
 160 int max_l=6,msp,sp,p,n,b,en,ds,st,y=0,mn,sh,uf,ln,hi=0,i,s
c,k,k0
 170 for i=0 to 13:o(i)=chr$(&HF4)+chr$(&H30+i):next
 180 dmy=chr$(&HF3)+chr$(&H20)
 190 ini=string$(max_l+4," ")
 200 for i=1 to 96-(max_l+4):ini=ini+chr$(&HF4)+chr$(62):next
 210 print "画面は大きいのがいい？";:s=inkey$:cls
 220 if s="y" or s="Y" then screen 0,0,1,0
 230 color[0,&HF818,&HDF30,&HFFFE]:color 10
 240 /*** 数字パターンの定義 ***/
 250 gj={&H7C,&H82C6,&HC6C6,&H8200,&H82C6,&HC6C6,&H827C,&H0}
 260 set_gaiji(&HF430,gj)
 270 gj={&H0,&H206,&H606,&H200,&H206,&H606,&H200,&H0}
 280 set_gaiji(&HF431,gj)
 290 gj={&H7C,&H206,&H606,&H27C,&H80C0,&HC0C0,&H807C,&H0}
 300 set_gaiji(&HF432,gj)
 310 gj={&H7C,&H206,&H606,&H27C,&H206,&H606,&H27C,&H0}
 320 set_gaiji(&HF433,gj)
 330 gj={&H0,&H82C6,&HC6C6,&H827C,&H206,&H606,&H200,&H0}
 340 set_gaiji(&HF434,gj)
 350 gj={&H7C,&H80C0,&HC0C0,&H807C,&H206,&H606,&H27C,&H0}
 360 set_gaiji(&HF435,gj)
 370 gj={&H7C,&H80C0,&HC0C0,&H807C,&H82C6,&HC6C6,&H827C,&H0}
 380 set_gaiji(&HF436,gj)
 390 gj={&H7C,&H82C6,&HC6C6,&H8200,&H206,&H606,&H200,&H0}
 400 set_gaiji(&HF437,gj)
 410 gj={&H7C,&H82C6,&HC6C6,&H827C,&H82C6,&HC6C6,&H827C,&H0}
 420 set_gaiji(&HF438,gj)
 430 gj={&H7C,&H82C6,&HC6C6,&H827C,&H206,&H606,&H27C,&H0}
 440 set_gaiji(&HF439,gj)
 450 gj={&H0,&H0,&H0,&H7C,&H82C6,&HC6C6,&H8200,&H0}
 460 set_gaiji(&HF43A,gj)
 470 gj={&H0,&H0,&H0,&H7C,&H0,&H0,&H0,&H0}
 480 set_gaiji(&HF43B,gj)
 490 gj={&H0,&H0,&H0,&H7C,&H0,&H0,&H7C,&H0}
 500 set_gaiji(&HF43C,gj)
 510 gj={&H7C,&H0,&H0,&H7C,&H0,&H0,&H7C,&H0}
 520 set_gaiji(&HF43D,gj)
 530 gj={-1,-1,-1,-1,-1,-1,-1,-1}
 540 set_gaiji(&HF43E,gj)
 550 /*** 効果音の設定 ***/
 560 m_init()
 570 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
 580 v={ 61,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
 590 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 600     31,   0,   0,  15,   0,  28,   0,   2,   0,   0,   0,
 610     31,   0,   0,  15,   0,  16,   0,   1,   0,   0,   0,
 620     31,   0,   0,  15,   0,  16,   0,   1,   0,   0,   0,
 630     31,   0,   0,  15,   0,  16,   0,   1,   0,   0,   0}
 640 m_vset(1,v)
 650 for i=1 to 7:m_alloc(i,100):m_assign(i,i):next
 660 m_trk(1,"@1 o6 q8 v12 d64")                /*start/erase
 670 m_trk(2,"@1 o5 q8 v12 c64")                /*miss shoot
 680 m_trk(3,"@1 o7 q8 v12 @L2dc+c>bb-aa-gg-fed+d")  /*UFO
 690 m_trk(4,"@1 o6 q7 v12 L16d8r>g8<d8r>g8<d8r>g8") /*over
 700 m_trk(5,"@1 o6 q7 v12 t106 d16q6d4d4d4d4 t120") /*clear
 710 m_trk(6,"@1 o5 q7 v12 g1")                 /*miss
 720 m_trk(7,"r1")                              /*wait
 730 /*** ゲームプログラムスタート ***/
 740 while 1
 750 msp=43:sc=0:mn=1:uf=0
 760 m_play(1)
 770 sh=3:b=30:en=16:ds=0:p=0:st=0:kh=string$(96," "):e=""
 780 locate 0,y,0:print ini
 790 locate 1,y,0:print string$(max_l-4," ");:num(right$("00000
"+str$(hi),6))
 800 m_play(7):while m_stat(7):endwhile
 810 locate 0,y,0:print ini
 820 locate 1,y,0:print string$(max_l-3," ");:num("16"):print o
(11);:num("30")
 830 m_play(7):while m_stat(7):endwhile
 840 while 1
 850 while inkey$(0)<>"":endwhile
 860 locate 0,y,0:print ini
 870 locate 1,y,0:print o(p);o(10+sh)
 880 sp=2:e="":ln=0:m=o(p)
 890 randomize(val(right$(time$,2)))
 900 /*** ゲームメインループ ***/
 910 while 1
 920 sp=sp-1
 930 if sp=0 then {              /*** 敵の出現 ***/
 940    if en then {
 950         if uf then { e=e+o(10):ln=ln+2:uf=uf-1
 960         } else {
 970         e=e+o(int(rnd()*10)):ln=ln+2:en=en-1
 980         }
 990    } else {
1000    e=e+dmy:ln=ln+2
1010    }
1020    if ln/2>max_l then sh=sh-1:p=100:break
1030    locate 3+max_l-ln/2,y,0:print e
1040    sp=msp
1050 }
1060 k=bitsns(7)
1070 if k<>k0 then {             /*** 自機の行動 ***/
1080    if k=8 then { p=p+1:if p>10 then p=0
1090    } else {
1100      if k=32 then {
1110        b=b-1
1120        n=instr(1,e,m)
1130        if n then {sc=sc+(max_l-ln/2+n/2)*10:e=left$(e,n-1)+
right$(e,ln-n-1):ln=ln-2
1140               if p<10 then {
1150                     locate 3,y,0:print string$(max_l," ")
1160                     m_play(1)
1170                     st=st+p:ds=ds+1
1180               } else {
1190                     sc=sc+300:locate 1,y,0:print string$(max_
l+2," ")
1200                     m_play(3):while m_stat(3):endwhile
1210                     locate 1,y,0:print m;c(10+sh)
1220               }
1230               locate 3+max_l-ln/2,y,0:print e
1240               if st and (st mod 10)=0 then st=0:uf=uf+1
1250          } else {
1260               m_play(2)
1270          }
1280      }
1290   }
1300 m=o(p):k0=k
1310 locate 1,y,0:print m
1320 }
1330 if b=0 or (en=0 and ds>=16) then break
1340 endwhile
1350 if en=0 and ds>=16 then {      /*** 1面クリア ***/
1360     locate 0,y,0:print ini
1370     locate 1,y,0:num(right$(kh+str$(mn),max_l-5)):print o(1
1);:num(right$("00000"+str$(sc),6))
1380     m_play(5):mn=mn+1:msp=msp-4:if msp<10 then msp=43:mn=1
1390     while m_stat(5):endwhile
1400     sh=3:b=30:en=16:d$=0:p=0:st=0:uf=0
1410 }
1420 if (en and b=0) or sh=0 then { /*** ゲームオーバー ***/
1430     locate 0,y,0:print ini
1440     locate 1,y,0:num(right$(kh+str$(mn),max_l-5)):print o(1
1);:num(right$("00000"+str$(sc),6))
1450     m_play(4):break
1460 }
1470 if p=100 then {                    /*** ミス ***/
1480     locate 0,y,0:print ini
1490     locate 1,y,0:print string$(max_l-4," ");:num(right$("00
000"+str$(sc),6))
1500     m_play(6):p=0:i=1
1510     repeat
1520     s=mid$(e,i,2):i=i+2
1530     if s<>dmy and s<>o(10) then en=en+1
1540     until i>ln
1550     while m_stat(6):endwhile
1560 }
1570 endwhile
1580 /*** リプレイするの ***/
1590 if hi<sc then hi=sc
1600 while inkey$<>"0":endwhile
1610 endwhile
1620 /*** 数字表示ファンクション ***/
1630 func num(a;str)
1640 str b
1650 for i=1 to len(a)
1660   b=mid$(a,i,1)
1670   if b>="0" and b<="9" then print chr$(&HF4)+b; else print
 b;
1680 next
1690 endfunc
```

リスト3

```
 10 dim int g(15)={
 20 &B1111111111111111,
 30 &B1000000000000001,
 40 &B1011111111111101,
 50 &B1010000000000101,
 60 &B1010111111110101,
 70 &B1010100000010101,
 80 &B1010101111010101,
 90 &B1010101001010101,
100 &B1010101001010101,
110 &B1010101111010101,
120 &B1010100000010101,
130 &B1010111111110101,
140 &B1010000000000101,
150 &B1011111111111101,
160 &B1000000000000001,
170 &B1111111111111111}
180 set_gaiji(&H7621,g)
```

Oh!XとOh!Xの読者の統計

Urakawa Hiroyuki 浦川 博之

さて，ここらで５年前のOh!Xと現在のOh!Xを比べながらアンケートハガキの集計になだれこみ，この本を読んでいる方々の姿を探ってみたりしたいと思います。はたして，あなたは平均的な読者なのか，それとも……。

Oh!MZがOh!Xと名前を変えて，はや５年だそうです。おめでとうございます。ありがとうございます。これもひとえに皆様の厚いご声援のおかげですが，ふたえにはスタッフが頑張ったからじゃないかななんて，ちょっと自慢だ，エッヘン。

しかし，アレですね，５年というと長そうで短く，実は長いというなかなかに味わい深い年月ですね。消費税はまだなかったし，青函トンネルもベイブリッジもできてなかったし，ヤクルトにはホーナーがいたし，第一，まだ昭和だったというのがすごい。かと思えば利根川教授はノーベル賞もらってたし，マドンナもマイケル・ジャクソンも日本に来てたし，「美味しんぼ」の「まったり」は知れわたってたし，スペースシャトルも爆発してた。ほらほら，だんだん古いような新しいようなわけがわからない状態になってきたでしょ。

というわけで，Oh!Xも５年前はどうだったか正確に覚えている人も少ないに違いないという仮説に基づき，ここでいっちょう昔のOh!Xと今のOh!Xを比べてみようというわけです。味方同士の新旧対決。仮面ライダー１号対２号。キカイダー対キカイダー01。このテの対決はたいてい新型のほうが性能が上なのに古いほうが勝っちゃうもんだけど，Oh!Xの場合はいかに？

データで調べるOh!X対決

赤コーナーァ，1987年12月号から1988年11月号のOh!Xゥ～(ぱちぱちぱち)。Oh!X改題最初の号から12冊が並んで入場です。背表紙は赤と茶色が基調。表紙はTadao Matsubaguchi氏によるイラストレーションです。表紙に「ポケコン」の文字が躍っているのが心強い。

青コーナーァ，1991年12月号から1992年11月号のOh!Xゥ～(ぱちぱちぱちぱち)。対する最新のOh!X12冊です。やや，こちらのほうが「ぱち」がひとつ多い。読者が増えたことを暗示しているのでありましょうか。白を基調にグリーンを織り交ぜたクールな背表紙で登場です。表紙はCGというあたりが貫禄を示しています。ソフトバンクのロゴマークがいいアクセントだ。

では注目の対決，いよいよ開始だ！

第１ラウンド　ページ数

バブルもバブルの崩壊も関係なくわりと同じ厚さで来ているOh!X。実際のページ数の変化はどうなんでしょうか。

では表１。おっと，1988年度のほうが1992年度のOh!Xを凌いでいる。いきなり新旧対決の定石どおりの展開だ。しかも１冊あたりの平均にすると差は10ページ。

続いて広告，これも1988年度のほうが多かったようです。カラーページでは1992年度のほうが多いんですが，1988年度の１色ページでぐんと差をつけられたようです。まあ，最近の不景気の影響も少なからずあるでしょう。

ちなみに1992年度でいちばん広告が多かったのは1991年12月号の37ページ。特集は音楽。「スターウォーズ」とか「出たな!!　ツインビー」が発売時期だった。

対する1988年度のほうは1988年10月号で38ページ。ゲーム特集の月だね。ぺらぺらと見てみると，スキャップトラストが「ガルフォース・怒濤のカオス」の広告を出しているのを発見。「ついに完成！　話題の超新星」というコピーを見て，なぜか熱い涙が溢れてきた私だった。

本文ページは1879対1842で1988年度のOh!Xの勝ち。ということは，昔に比べて今のOh!Xは，「記事ページ数は少し減って全体的に薄くなった」ということになる。昔のOh!Xのほうが内容があったというわけか？　いやーん，定価は60円も上がってるのに。こりゃ今の読者さんから怒られそう。

第１ラウンドは1988年度のKO勝ち

だが待てよ。リストの分量を見ると，1988年度の718ページから1992年度は377ページに大幅ダウンしている。１冊あたりにするとその差は28.4ページだ。それなのに記事全体のページが３ページしか減ってないってことは，読者が読むページは，実はすごく増えてるわけだよなあ。

さらに，1992年度には２度の付録ディスクがついている。このディスクに入ってるプログラムを，もし全部リストにして掲載してたら絶対1988年度のページ数は軽く追い抜いてるわけで，ということはひょっとすると1992年度のほうが記事もプログラムも増えているってことになるんじゃないだろうか。

おっっとぉ，新型が負けたかに思われたが，奇跡の復活で逆転勝ちを収めるという，またまたこのテの対決にありがちな展開で決着だあ！

第１ラウンドは物言いがついて，行司差し違えで1992年度の勝ち。決まり手はディスク出し投げ。

続いて，第２ラウンド。

第２ラウンド　読者の年齢層

一説によると，Oh!Xの読者というのはいったん買い始めると，ずーっと読み続けて

表１　本誌のページ数

	総ページ数	広告	記事	リスト
1988年度	2304	425	1879	718
月別平均	(192.0)	(35.4)	(156.6)	(59.8)
1992年度	2176	334	1842	377
月別平均	(181.3)	(27.8)	(153.5)	(31.4)

くれる方が多いとか。ありがたいことです，ホントに。ということはですね，1年間同じ読者の方々が買い続けてくれると，読者の平均年齢は1歳上がってしまうわけですね（統計学に詳しい人は黙ってるように）。名づけて「Oh!Xジジイ化現象」。これを下げるには，それより若い人が新しく読み始めなければいけないと。ジジイになった人が読むのをやめても下がるんですけど，ダメですよやめちゃ。

そのへんを踏まえて，図1を見ていただきましょうか。STUDIO Xとハミダシに掲載されていた方々の平均年齢です。1988年度と1992年度では思ったほど変わってませんね。4年かかってちょうど1歳上がっただけ。ハミダシのほうも0.7歳しか上がってない。意外にOh!Xの読者の高齢化は進んでいなかったみたい。

第2ラウンドは両者引き分け

どちらの年度もハミダシのほうが平均年齢が若干低いってのは面白い結果ですね。近況報告や編集室への意見が多いSTUDIO Xに比べると，ハミダシは一発ギャグが載る可能性が高いからなあ。そのテのハガキを送ってくる人はどうしたって若い人（ああっ，こんな言葉を使うようじゃ私もジジイの仲間入りだぁ）になっちゃうわけだ。

ちなみに，作ってるスタッフのほうですが，これは確実に平均年齢は上昇しているでしょう。だって，編集室をずーっと見渡すと，結構前から見た顔ばっかり。それでも4年前はスタッフじゃなくて読者だったっていう人は多いから，4年間で4歳上昇ってことはないと思いますけどね。

ともかく，読者の平均年齢は4年間で約1歳の上昇という結果になったわけですが。しかし，これはあくまで「STUDIO Xとハミダシに採用された人」の平均年齢だから，安心はできない。面白いハガキを書いてくる人ってのはやっぱりこれくらいの年齢の人だと思うし。

第一，掲載されてる人々が読者全体を正しく反映してるとしたら，ウチの読者は受験生と浪人だらけだ，なんてね。

考えてみれば，Oh!Xの正しい読者像ってどんなんでしょう？　X68000ユーザーで浪人でローディストでプログラミングができて少女マンガを読む？　うーん，それは読者全体の実態とはちょっと違うような気が……。どっちかっつーと「あるべき読者像」かなぁ（この発言も問題アリ）。

読者分析といえば6月号で荻窪圭氏が読者アンケートの分析をやってたけど，アンケートに答えてくれる人も結構パワーのある読者の方々だから，これまた全体像とは遠いかも。読者ハガキのほうがまだ……。

じゃ，それらの再確認のために最近のアンケートハガキの集計でもしてみましょうか。Oh!Xでは編集者を始め，スタッフもほとんどのアンケートハガキに目を通しているんですが，平均とか回答内容のランキングまでは集計してないんですね。こいつをちょっと調べたら，Oh!Xの読者像がわかるかもしれない。読者のあなたも，ほかの人たちが何を考えてるか，ちょっと気になるでしょ？

Oh!X生活総合研究所年次報告

というわけで，Oh!X生活総合研究所へようこそ。ここでは，毎月定期的に送られてくるアンケートハガキを統計学的に集計し，認知心理学的にまとめたうえで，享楽的かつ刹那的な考察を加えようという，場当たり的かつお気楽的な試みをしてみたいと思います。

対象にしたのは1992年10月号のアンケートハガキ500枚（任意抽出）。編集室がいま手にしているいちばん新しいデータです。あなたが先々月に出したあのハガキもこの中に入っているかもしれない。

ではさっそく見ていくことにしましょうか。私も早く見たい。

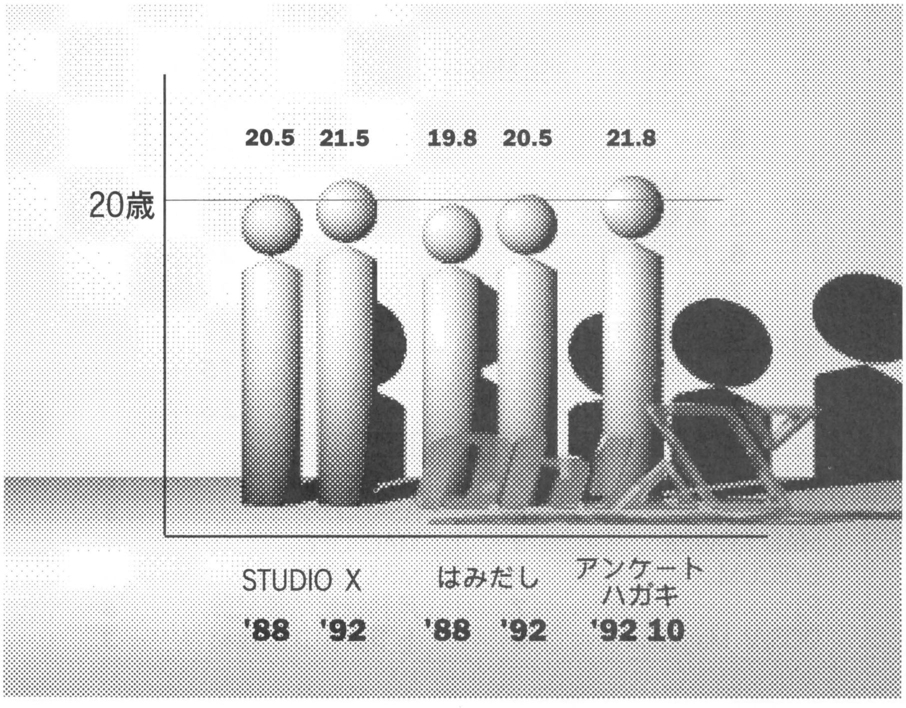
1）　平均年齢

21.8歳。ということは，1992年度のSTUDIO Xの平均とほぼ同じ。STUDIO Xに取り上げられている人は，少なくとも年齢的には，読者の中のごく普通の人でしたね。意外といえば意外だな。この年だと大学生なら3年か4年。専門学校出身だと社会人1年目か2年目ってとこですね。Oh!Xの場合は浪人が多い（ような気がする）ので，大学4年といい切れないところが哀愁をそそります。「受験雑誌Oh!X」というわりには，受験期を過ぎちゃった方が多いようで。

2）　職業

この平均年齢の予測を検証するために，職業のデータを引っ張ってきました（図2）。社会人を筆頭に，大学生，高校生の順とな。大学生がいちばん多いかなと私は思ってたんですけど，予測が外れたな。実際

図1　読者の平均年齢



には“21歳・会社員”なんて人も多いから，ハガキ全体のノリが大学生っぽく見えちゃうんでしょうけども。

そのほかには主婦やフリーターが入ってるわけですが，不明の数が結構多い。書きたくないのか，ホントに身分不明なのか知りませんが，宇宙人でも魔法使いでも西川善司でも職業はちゃんと書いてくれるとうれしいんだけど。

Oh!Xとしては気になるのが浪人の数ですが，心配されたほど多くないようなので，受験生の皆さん，安心してOh!Xに読みふけってください。なに？　いっそ浪人ばかりのほうが落ちたときが安心だ？

図2　読者の職業別割合

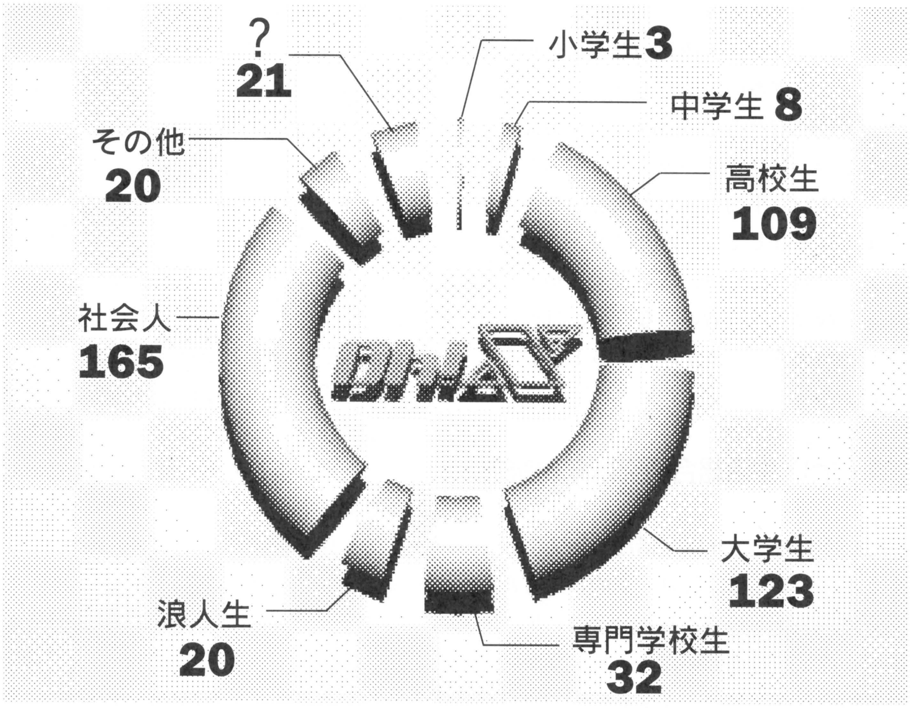


図3　併読誌ベスト10およびその他

1.	マイコンBASIC Magazine	99
2.	LOGIN	90
3.	ASCII	56
4.	C Magazine	45
5.	電脳倶楽部	42
6.	POPCOM	29
7.	I/O	28
8.	コンプティーク	17
9.	バックアップ活用テクニック	15
10.	テクノポリス	13

Oh!PC	9	My Computer Magazine	5
Oh!FM TOWNS	8	ASAHIパソコン	5
MAC POWER	8	MAC LIFE	4
EYECOM	7	MSX FAN	4
PJ	7	月刊情報処理試験	3
DOS/V Magazine	6	トランジスタ技術	3

3)　併読している雑誌

図3のとおり。トップはマイコンBASICマガジン。あのログインを抑えて併読誌の第1位に輝くとは立派。Oh!Xの読者は，やっぱりプログラムリストが載っている雑誌に弱いんでしょうかね。それとも山下章のファンが多いせいだったりして。

僅差で2位につけたのがログイン。ゲーム関連の記事が充実していて，プログラムと活用記事中心のOh!Xとはいい補完関係，ラーメンにギョウザ，ケンタッキーに焼きむすびっていう気もします。いや，Oh!Xも負けないように頑張るけどね。1，2位がこれだけ他を引き離してるのは，読者がいかにゲームが好きかというのをよく示している結果ですね。

3位には一転してお堅い(?)アスキー。OS関連やコンピュータビジネスに興味のある読者も少なくないってことですな。勝手な推測だけど，アスキーを併読してる人って，ふだんはX68000以外のパソコンを使ってて「X68000は趣味ですから」とかなんとかいってそう。パソコン通信にも興味があって，フリーウェアに詳しいかな。

上位3誌はメジャーな雑誌が入ったのに比べ，4位と5位はなかなか味のあるヤツらがきています。C MAGAZINEでは現在X68000のC言語プログラミングの連載をやっているので，併読する人が多いのもわからなくはないですが，やはりこれを併読誌に挙げてる人はかなりのパワーユーザーじゃないかな。

5位。 Oh!X読者なら知らぬ者はないという電脳倶楽部。いわずと知れたあの広告主であります。ちなみに，満開の電子ちゃんはOh!Xの記事ではありません。「いちばんよかった記事」に書いてこないように。満開製作所の人は喜ぶかもしんないけど。やはりそのホレ，電源オンですぐ起動だし，マウスひとつで楽々操作というところが5位獲得の原因でしょうか。

えー，以下の順位は別表を見ていただきたいんですが，全体の傾向としては「ゲーム2誌に人気集中」「パソコン通信誌の票が少ない」「家庭用ゲーム機の雑誌も弱い」といったところです。

4)　所有機種

500人から統計を取ったら，マシンの所有台数合計は678台にもなってしまいました（図４）。１人平均1.36台ですか。日本人の出生率が1.52だから子供よりは数が少ないねって，そういう問題じゃない。

その所有台数678台のうち，X68000はなんと455台。X68000を複数所有している人もいるから（これがホントに結構いる）単純にはいえないけれど，読者の間の普及率は実に91%。高いと見るか低いと見るかは難しいところではありますが。

X1またはMZのユーザーであると回答した人も，大多数はX68000も持っていて，X1やMZだけを持っているという人はかなり少ないですね。やはりこれも時代の流れ。

X68000を機種別に見てみましょう。いちばん多いのがACE。次がEXPERTと。このへんの順位は６月のアンケート集計と大差なし。XVIはもっと所有比率が高いかと思ったけど，６月以来あんまり伸びてませんね。票数としてはほかの機種と並ぶ程度。

それにしてもCompactの５票ってのは……。いくら3.5インチとはいえ，MZ-1500ユーザーと同数ってのはちょっとないんじゃないの。Compactはコンピュータ界のアルシオーネになってしまうのか。シャープさん，なんとかしてね。

X1，MZの内訳を書いておきましょう。

単一機種でいちばん多かったのがX1 turboのModel30。次がturboZですから，turboの勢力も侮れないですな。なにせZだけでCompactの４倍の勢力がある（禁句）。MZのほうではMZ-700のユーザーがいちばん多いという意外な結果。古籏氏による強力な活動がユーザーをつなぎとめているのか？　MZ-2500ユーザーが６人ってのは少ないような気がするな。いいマシンだし，投稿も来てないわけじゃないのに。

それにしてもX1turbo Model10ユーザーの１人とX1Dのユーザーも３人。X1Dとかはメディアの入手も困難でしょうが，がんばっているようですね。

そのほかのマシンや，ナイコンでありながら読んでくれている方々は68人。ありがたやありがたや。これもよく見ればXVIユーザーと同じくらいいるわけで，やはりこの，グローバルで普遍的なOh!Xのムードが受けたんではないかと（ウソ度70%）。

5）パソコン歴

お世辞にも初心者向けとはいえないこの雑誌，読者のパソコン歴もけっこう長いのではないかと推測されます。というわけで図５をご覧ください。

平均は断定できませんが，おそらく５年から６年の間であると推測できます。初めてのパソコンがX68000だったという人も半分ぐらいいると。もはや「データレコーダの真似」という宴会芸は通用しない。

この分布のどれくらいまでが「自分は初心者」と思ってるんでしょうね。２〜３年のうちはそう思ってる人がほとんどだと思いますが，もしそうだとすると読者全体に占める初心者の割合もそんなに低くはない

図4．Oh！X読者の機種別所有者数（1992年10月号アンケートハガキより）

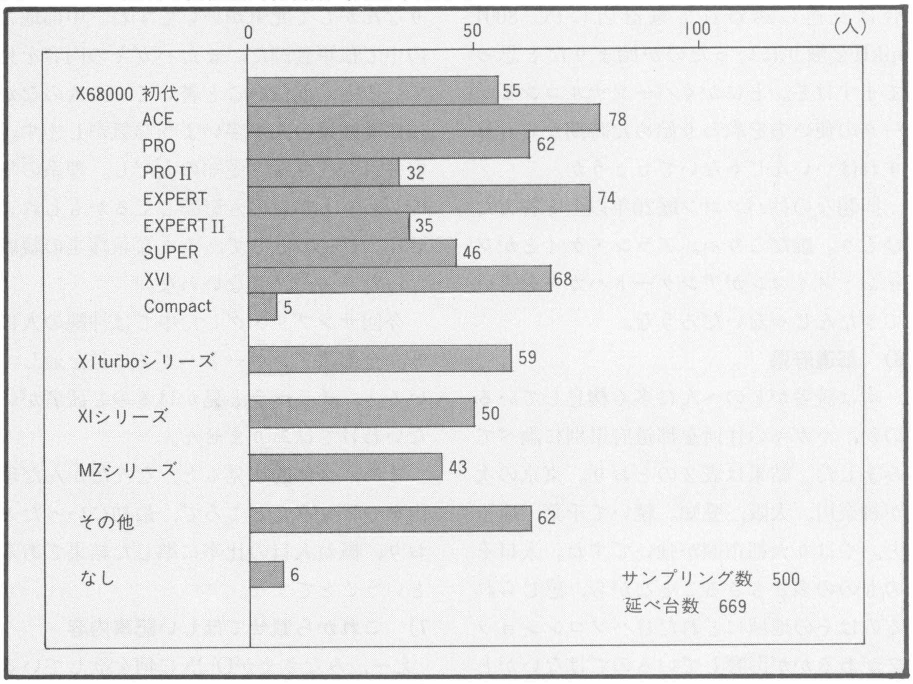
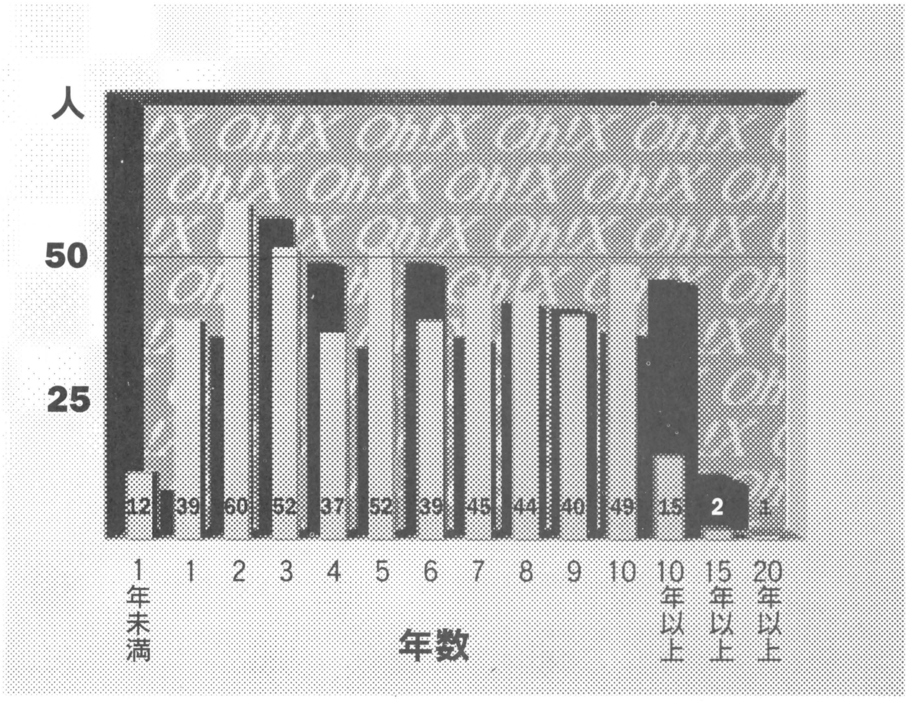


図5　読者のパソコン歴



ということになります。ハガキを個別に見ているとX68000を買ってパソコンを始めたという人もわりといるようですしね。そういうのにかぎって高校生だったりして，まったく憎たら……，いや，うらやましい。

また,「どのへんをパソコン歴の始まりにすればいいんだ」というハガキをたまに見かけます。ファミコンを買ったのは，パソコン歴の中に入れていいのかとか。私の場合は友達に誘われて電器店にPC-8001mkIIを触りにいったのが始まりだと思ってますけど。とにかくパーソナルコンピュータの使い方を教わり始めた時期から計算すればいいんじゃないでしょうか。

問題なのはパソコン歴20年以上と答えたひとり。誰だこりゃ。アラン・ケイとかフォン・ノイマンがアンケートハガキを書いてきたんじゃないだろうな。

6） 都道府県

次は読者がどのへんに多く棲息しているのか，ハガキの住所を都道府県別に調べてみました。結果は表2のとおり。東京の次が神奈川，大阪，愛知。続いて千葉，埼玉と。やはり大都市圏が強いですね。人口そのものの多さもさることながら，感じられるのはその地域にどれだけパソコンショップがあるかが影響しているのではないかということです。

それを示しているのが北海道。札幌の電気街が幸いしたのか，関東近県に迫る27人を記録しています。もともと北海道はソフトハウスも多いですしね。ズームとかハドソンソフトとか，デービーソフトにマイクロネット。それにアレかな，寒いから室内の娯楽が盛んなのかもしれないな。あ，でもそれは東北地方の他県を見てみると当たっていない。

愛知県も大阪を相手に対等に渡り合ったりなんかして元気がいいですね。中部地方の中で孤軍奮闘だ。またハガキの内容を見ていると，面白いこと書いてくる人のなかには愛知県の人が多いような気がします。安井百合江さんも愛知の人だし。理系の学校がわりと多いのも影響してるかもしれません。ひょっとして，みんな荻窪圭の親戚だったり……，しないわな。

今回サンプリングした中では沖縄の人は0。全部のアンケートハガキに目を通していると，チラホラと見かけるので読者がいないわけではありません。

まあ，全体的に見ると，なんだかんだ理由をつけてみたところで，最初にいったとおり，概ね人口の比率に準じた結果であるということですな。

7） これから載せてほしい記事内容

えー，みなさまがOh!Xに何を欲しているかがいちばんよくわかるコーナーです。私も特にこの結果が気になります。

表3を見てください。1位はグラフィック関連記事。やはりX68000ユーザーのこの分野に対する興味は強い。実際DōGAなどへの取り組みも積極的ですしね。

2位はプログラミングと。5位のC言語というのも同じ系統かな。やはりグラフィックにしても音楽にしても，プログラムというものを理解していることがX68000の場合重要ですから，このテの要望が絶えないのもわかります。

3位のMIDIですけど,次点の音楽関係と一緒にするとトップに躍り出ちゃうほどの強さ。AV機能の活用に対する感心は並々ならぬものがあるようです。

同時に3位になったのが,「X68000の今後」。X68000もいよいよ6年目ですしね。そろそろアーキテクチャの変更があってもおかしくない。X68000がこの先どうなるのかは私も興味があるところです。とかいってはぐらかしたりして。

あとはゲーム，新作ソフト，これはTHE SOFTOUCHのさらなる充実を求めているってことかな，ハードウェア関連，BASIC(！)，パソコン通信の順。さすがX68000，AV機能の活用とプログラミングが際だっている。加えてX68000の今後について知りたいっていうんだから，X68000が何かの手段になっているのではなくそれ自体が目的であることが見てとれます。

こうして回答の種類をみると,「載せてほしい記事内容」=「これから組んでほしい特集」と置き換えて回答してくる人が多いみたいですね。

表2 都道府県別読者数

北海道・東北							
北海道	27	青森	5	岩手	2	宮城	12
秋田	2	山形	3	福島	10		
関東							
茨城	11	栃木	10	群馬	12	埼玉	28
東京	51	神奈川	45	山梨	1		
信越・北陸							
長野	4	新潟	5	富山	6	石川	7
福井	4						
東海							
岐阜	6	静岡	10	愛知	34	三重	8
近畿							
滋賀	8	京都	14	大阪	39	兵庫	14
奈良	6	和歌山	3				
中国							
鳥取	1	島根	2	岡山	8	広島	10
山口	6						
四国							
徳島	1	香川	7	愛媛	9	高知	2
九州・沖縄							
福岡	16	佐賀	2	長崎	1	熊本	8
大分	3	宮崎	1	鹿児島	3	沖縄	0

表3 これから載せてほしい記事ベスト10

1.	グラフィック	29
2.	プログラミング	28
3.	MIDI	21
4.	X68000の今後	21
5.	C言語	19
6.	ゲーム	18
7.	新作ソフト	17
8.	ハードウェア	16
9.	BASIC	15
10.	パソコン通信	14
次点	音楽	13

まとめと展望

Oh!Xの読者って，データを見れば見るほど正体不明になってくるような気がする。C MAGAZINEを読んでる人もいれば，ゲームの記事を充実させてほしいと願う人もいる。38歳のオジサンもいれば，19歳の女子大生もいる。内容が簡単すぎると怒る人もいれば，難しすぎてさっぱりわからないと悲しむ人もいる。

ま，しかし，ひとついえることはですね，みんなX&MZシリーズが好きで，みんなグレイトなユーザーになってやるぜという意気込みをもっている人ばかりだということですね。コンピュータに向き合っていることが楽しいという気持ちをあなたがもっているかぎり，Oh!Xはあなたの味方ですよとカッコいい見栄をきったところで一件落着。この次をお楽しみに（この次って何なんだろう……)。

WE WANT YOU!

Oh!Xの掲載記事を理解するうえで重要となるキーワードに「パーソナルコンピューティング」という言葉があります。なにも，難しい概念などではありません。Oh!Xが提唱しているのは，「パーソナルコンピュータをちゃんとパーソナルコンピュータとして使う」，というごく単純なことにすぎないのです。

それぞれの人がそれぞれのスタイルでパーソナルコンピューティングを楽しんでいると思います。それがどんなものであるかを知ることは，本誌の誌面作りにとって非常に重要なことなのです。そして，Oh!Xが発信したメッセージを皆さんが受け取り，それに対する皆さんのメッセージが今後のOh!Xの方向を決めていくことにもなります。

実際，Oh!Xの誌面はスタッフだけが作っているものではありません。これまでのOh!MZ/Xの軌跡をたどると要所要所で読者投稿作品が大きな影響力を及ぼしていることがわかります。読者の力がこれまでのOh!Xを支えていたといっても過言ではないでしょう。

しかし，影響を与えられているのは投稿作品だけではありません。実はそれ以上の影響力を持つのがアンケートハガキによるメッセージです。Oh!Xの全体的な方向性を決めているのは誌面にはあまり現れない多くの人の意見なのです。読者層が変われば記事が変わる，というほど単純なものでもありませんが，記事の方向性に多大の影響を及ぼしています。

投稿作品はそれ自体が強いメッセージでもあります。強いメッセージは歓迎します。また，アンケートハガキの回収にもご協力ください。多くの方の意見が揃ってこそ，よりよいフィードバックが行われます。

私たちはいつでも皆さんからのメッセージを求めています。

〈重点募集項目〉

- **●オリジナル音楽データ**
- **●カードゲーム**
- **●SX-WINDOWアクセサリ**
- **●Z's-EX用外部コマンド**

イラスト投稿の規定

サイズはハガキ大（A6判）以上であれば可。B5判くらいまでは可能ですが，取り扱いの手間や現実的な問題としてハガキ大を一応の標準とします。いずれにせよ，掲載時にはかなり縮小されることを考慮して描いてください。

一応の推奨形式は以下のとおりです。

1）　ハガキ大のケント紙で郵送

ハガキでも結構ですが，たまに裏面にも消印が押される場合があります。

2）　黒1色（薄ズミ不可）

墨汁は汚れの原因になることがあります。製図用インクがおすすめです。原稿は縮小されますのでスクリーントーンの80，90番台（レトラセットの場合）などや色の濃すぎるものについては再現は保証されません。残念ながら，カラー原稿はごくたまにしか掲載されません。

内容に関して特に規制はありませんが，時期もの（正月，クリスマス，季節もの）などについては，掲載が予想される時期を考慮して早めに送ったほうが有利になることがあります（年賀状は例外）。

それでは，皆さんの力作をお待ちしています。

協力スタッフ募集

Oh!Xでは誌面作りに参加していただく協力スタッフを募集しています。

スタッフとして活動する熱意があり，東京近郊にお住まいの方でソフトバンクまで来社可能な方。特に時間的な束縛はありませんが，ある程度時間的な余裕がある方に限ります。基本的に学生を対象としていますが，十分に時間的余裕と余力があれば社会人も可とします。ただし，18歳未満の学生および浪人生の方については採用予定はありません。

応募要項です。ライター希望の方はOh!X誌面2ページ分相当（2000字程度）の自由論文に自己紹介文を添えて「Oh!Xスタッフ希望」係までお送りください。

また，文章力には自信がないけどプログラムなら……という方でも技術スタッフとして，参加していただく場合があります。こちらを希望の方は自由論文の代わりに，これまでに制作した自作プログラムとその解説などを一緒に応募してください。

書類選考後，採用者の方にはこちらから連絡いたします。

投稿大募集

Oh!Xでは読者の皆さんによる投稿作品を常時募集しています。

未発表の作品であれば，グラフィック，音楽，システムプログラム，ツール，ゲーム，ハードウェアなどジャンルを問いません。数当てゲームからOSまでなんでも受け付けています。機種についても（メーカー，年代など）特に限定はしませんが，雑誌の性格上扱いにくい場合もあります。

誌面に載りきらない大きなアプリケーションなどはディスクメディアを使って配布することが考えられます。その形態のひとつはご存じ付録ディスク，そしてもうひとつは別冊形式によるものです（近日発売予定のZ-MUSICシステムに続き，今後もいくつかのOh!X MOOKシリーズが予定されています）。

また，特に掲載されることを目的とせず，「こんなものを作ってみました」といったプログラムでもかまいません。気軽に作品を送ってみませんか。

投稿募集要項

1）　お送りいただくプログラムには，住所，氏名，年齢，職業，連絡先電話番号，機種名，使用言語，動作に必要な周辺機器，マイコン歴などを明記のうえ，封書の宛先の最後には「Oh!X LIVE」，「全機種共通システム」，「投稿ゲームプログラム」など，プログラムの内容を明確にご記入ください。

2）　投稿されるプログラムには詳しい内容を記入した原稿を同梱してください。ディスクの中にドキュメントファイルの形式でのみ記述している方がいますが，郵送時の事故などでメディアが破壊されることもありますので，必ず文書を添えるようにしてください。一緒に変数表，メモリマップ，参考文献などがあればなお結構です。また，掲載に際してお送りいただいたプログラムやデータ原稿については，当方で加筆修正をさせていただくことがあります。

3）　お送りいただくプログラムは事故防止のため最低2回はセーブしておいてください。基本的に同封されたフロッピーディスク，カセットテープ，クイックディスク，原稿などについてはご返送いたしませんので，あらかじめご了承ください。

4）　ハード製作関係の投稿につきましては，最初は内容のわかる原稿のみお送りいただければ結構です。その後，当方で製作物が必要だと判断した場合には改めて連絡いたします。

5）　お送りいただいた作品の採用につきましては，掲載号が決定した時点で当方より連絡いたします。特にツール関係，ハード関係などのものにつきましては特集内容などを考慮したうえで採用決定されますので，結果を連絡するまでに時間がかかる場合があります。

6）　投稿いただいたプログラムにバグなどが発見された場合は新しいプログラムの入ったメディアと一緒に文書にてご連絡ください。

7）　掲載されたプログラムに対しては当社規定の原稿料をお支払いたします。また，投稿されたプログラムの著作権などはすべて制作者に保留されますが，いわゆる「PDSなどとしてネットにアップする」ことなどを希望される場合には必ず事前に編集部までご連絡ください。なお，一般的モラルとして，他誌との二重投稿または，他誌に掲載されたプログラムの移植などについては固くお断りいたします。

その他，不明点については編集室まで問い合わせてください。

宛先
〒108　東京都港区高輪2-19-13　NS高輪ビル
ソフトバンク株式会社
Oh!X編集室「投稿プログラム」係

キャラグラゲームのススメ

シュールな風景

Shibata Atsushi　柴田 淳

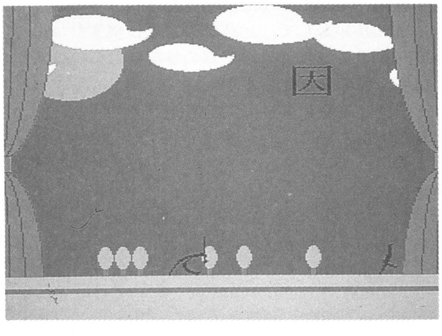
いつもオリジナリティあふれるゲームを作る柴田君の意欲的実験作。今回はX68000でキャラグラ(?)を駆使したゲーム風デモソフトだ。優雅に動くチャイニーズキャラクタたちに注目だ。

ゲームの面白さに対する論議は絶えない。ゲーム性がどうのとか感情移入がこうのとか，そのような話をよく目にする。ただ巷によく見かけるその種の話というのは，出来上がったゲームを取り上げてこれはすごいとかなんとかいっている場合が多い。

ゲームを語るうえで，いわゆる評論の手法はかなり確立されてきているようだ。そのようなシゴトはもちろん大きなことだと思うのだが，本当はもっと，出来合いのものを編纂する理論から発展して，それを作り手の理論に昇華させる動きがあってしかるべきだと，僕は思うのだ。

とはいえ，作り手の側の理論，いい換えると面白いゲームを作るための方法論を確立するのはたやすいことではない。だいたい，どんなゲームが面白くてどんなゲームが面白くないかを一般的に定義することからして，かなり難しいシゴトである。あるゲームを面白いという人もいれば面白くないという人もいて，つまり人間の趣向というのには個人差があるわけで，そこまでひっくるめたカタチの，大統一理論を打ち立てるのは並大抵のことではないのだ。

ところで，あくまでも主観的な判断の域を出ないが，僕はある方法でゲームの面白さを判断している。面白いゲームをやると，僕は決まって涙が出るのだ。



このプレイ中の涙は，ゲームのデキを判断するにはなかなかよい情報を提供してくれるのだが，困ることが2つある。第一に涙で視界が曇って画面がよく見えない。拭おうとすれば当然スティックなりトリガなりから手を離さなければならなく，シューティングなどではそのために窮地に追い込まれることもままある。次に困ることは，周囲からいかにも「不可解だ」といわんばかりの視線を浴びることである。まあ，ゲームセンターで泣きながらゲームをしている男を見れば，誰でもギョッとする。それがわかるからかえって周りの視線が気になってしまい，集中して遊べなくなる。せっかく出会った面白いゲームなのに，十分に楽しむことができないのだ。

でもいったいどうした理由で，ゲームをすると涙が出てくるのだろうか。涙腺が極度にユルいという特異体質なのだろうか，それとも深層心理が微妙に影響して，涙を流させるのだろうか。このことは僕にとって永年の謎で，ずっと悩んでいたのだが，先日ひょんなことから謎が解けた。僕はゲームをしており，それがまた面白くて，当然目からは涙がチョチョ切れていた。ゲームが終わり椅子から立ち上がると，隣で見ていた友人が，

「おまえゲーム中，ずっと瞬きしてなかったぞ」

といったのである。なるほど，涙も出るわな，瞬きしなけりゃ。

面白いゲームを前にして瞬きもできず，ドウドウと涙を流しているときの自分を冷静に分析してみると，なかなか滑稽で自分のことながら笑ってしまうのだが，しかしここで，いったいどうして瞬きしないんだという問題にブチ当たる。いや逆にいえば，瞬きができないほど，つまり目が離せないほどゲームが面白いのではないか。次から次へと場面が変わり，息もつかせぬ演出で，プレイヤーの目を離さない。シューティングであれアクションであれ，そのようなゲームがあったらやっぱりやってみたい。

ダラダラしてきたので結論を急ぐと，僕はゲームの面白さはやはり，そこに注ぎ込まれる情報量で決まると思う。絶えず情報を受け取れる状況に置かれると，どうしてかサル的部分がよみがえって，人間はなす術もなく恍惚とするばかりなのである。なんか表現がエッチな気もするが，面白いゲームとはそういったものではないか。

情報量と共感の関係

しかし，この情報量という判断基準も実は万能ではない。世の中にはテトリスのような，注ぎ込まれている情報量が極度に少なく，それでいて面白いゲームが存在するのだ。ただし，あのゲームはシステムの中に「組み合わせ」を導入していて，それが見かけ上の情報量を増やしているのだが，それでもまだ最近のゲームに比べて情報量は少なめである。

さて，ここでゲームの面白さを数値で表すことを考えてみよう。ゲームの面白さは情報量で決まる，つまり面白さを表す数値をＦとすると，Ｆは情報量を示す値Ｉに比例する関数として表せる。で，ゲームによっては，その情報量が面白さＦに与える影響が上方にシフトしていたり，あるいは下方にシフトしている場合もありうる。つまり，状況によって関数の傾きが変わるのである。その傾きをＳとすると，ゲームの面白さの数値には，

$F = S \times f(I)$

といった関数で得られそうである。要するにゲームの面白さには，情報量のほかにSという係数が関係しているのだ。

では，このSという係数はいったいなんであろうか。おそらくこれは，情報の量でない部分，つまり「質」であると思う。あるいは，その場その場における情報の的確さとでもいえようか。たとえば科学に興味を持っている人にとっては，SF小説というのは有意義な情報源であるわけだが，科学に疎い人にとっては，SFなんてただ難しいことが書いてあるだけで，読むに値しないものであろう。人にはそれぞれ趣向というものがあって，たとえ膨大な情報量を持ったものであろうとも，その趣向に合致しないと情報の重要性が著しく減少する。人によって面白いと感じるゲームがマチマチであるのは，このようなことから説明できないか。

情報量にかかる係数が質であるとしたら，qualityの頭文字を取ってQとでもすべきだろうが，ここではあえてSとした。それにはもちろん理由があって，というのは情報の質というのは，究極的にはプレイヤーが感じる「共感(sympathy)」であると思うからだ。あるいは感情移入という耳慣れた用語で置き換えてもいいかもしれないが，僕としては共感と呼んだほうがしっくりするし，用語的にも的を射ているはずである。

面白いゲームを作ろうと思えば，まずその中にできるだけの情報を詰め込んで，なおかつその情報の配列をプレイヤーの共感を得るに足るようにすればいい。しかし，情報というのはいってみれば地道な作業の反復でいくらでも増やせるのであり，すると後半部分が大きな問題となってくる。徹底的な市場調査をして，プレイヤー層を完全に把握するといった方法で，あるいは気まぐれな消費者の指向性に合致する情報を見つけることができるかもしれない。が，もっと手っとり早く，いつでも誰にでも，共感を与えうるような情報があればいいわけだ。

それが，なにを隠そう「斬新なアイデア」であると，僕は思うのだ。

なんかわけ知り顔でこんなことをいうと，表向きはすごいことをいっているような気がするかもしれないけど，それはあくまでも表面上のことにすぎない。だって，僕は「ゲームはアイデアだ」と，ごくごく当たり前のことをいっているだけだからである。これじゃやっぱり空しいので，今度はこの「アイデア」について，もう少し掘り下げてみることにする。

アイデアの泉

アイデアというのは，いったいどうやって生まれるのか。一般的には，なにか超自然的な霊感みたいなものが働いて，その瞬間にアイデアが生まれる，とでも信じられているのだろう。インスピレーションがまったく関係していないとはいい切れないが，それでもアイデアを生み出す過程のうち，かなりの部分は単純な，論理的な操作に還元される。それはテクニックであり，つまり誰でも一定の訓練をすれば身につけられるものなのである。

たとえば，ゲームのアイデアを練る場合を考えてみようか。ここで「ゲームのアイデア」という条件を設定することによって，さっそく選択可能な内容が限定される。つまりたくさんのアイデアのうち，ゲームにできないようなアイデアが思考の外に追いやられるのだ。で，次に「コンピュータのゲーム」という条件を設定すると，「表示は2Dのモニタ上」だとか「入力はジョイスティックから」といったふうに，どんどん新しい条件が導き出される。

このようにできるだけたくさんの条件を設定することによって，だんだんと選択肢が減っていく。そしてすべての条件が出揃ったときに，ふさわしいアイデアが浮かび上がるのだ。とするとアイデアは「生み出す」というより「絞り出す」もののように思える。アイデア生成の過程というのはほとんどが論理的な絞り込みの作業であり，そこを支配するのは霊感でもなければ直感でもないのである。

アイデアは決して「無」からは生まれない。アイデアの泉は，いってみれば「混沌」のようなものである。いい換えればアイデアの「ない」状態というのは「すべてある」状態で，また逆にいえば「すべてある」混沌は無に等しく，そこから設定した条件に合致する内容を抽出して初めて，モノゴトは意味を付加されるのである。

アイデアの発想法について一般的に信じられていることは，僕にはほとんどが間違いであるように思われる。第一に，アイデアとインスピレーションの関係についての一般的常識が間違いであるのはいま示したとおりだ。必要なのは先天的な霊感よりむしろ，論理的な思考能力である。斬新なアイデアも，いまいった論理的な操作で出てくる。最後に「既存のものを取り除く」という条件を打ち立てるだけで，新しいアイデアの候補が浮かび上がるという寸法である。

ただ，アイデアを絞り込む過程において，変な常識にとらわれてしまって，たとえばとっぴなアイデアを削ってしまったりするのは逆効果だろう。そのようななかにこそ斬新なアイデアが含まれている場合が多く，余計なことは考えず，完全に論理に従う，という鉄則を忘れてはならない。

キャラグラの必然性

また，巷ではいいアイデアを生み出すためには視野を広げることが大切と思われているようだが，これも僕にいわせればまったく逆である。いっそ逆に厳しい条件を課して，アイデアの範囲を思いっきり狭めてしまうのだ。そしてそこからいかに抜け出すか，狭められた状況をいかにして広げるかに全精力を注ぎ込むことによって，アイデアになんともいえない「うま味」みたいなものが加わる。

ここでX68000のゲームの周辺に話を移す。いわずと知れたことだが，X68000にはスプライトだとかBGだとか，ゲームを作るにはたいへん重宝な機能が備わっている。で，たとえばシューティングゲームを作ろうとすると，スプライトでキャラクタを描いて，あとはBGをスクロールさせれば，「てきとーにゲームらしいゲーム」は出来上がってしまうのだが，この「てきとーに云々」というのが非常に問題である。

このてきとーなゲームというのは，おそらく自機や敵機のサイズはすべて16×16ドットだろうし，多重スクロールしている背景も，なんか8×8の格子が透けて見えそうな代物であるに違いない。こういうゲームというのは，ひと言でいい表すとナンジャクなのである。所詮ハードは書き換えのきかない「固い」機構にすぎないのだ。スプ

ライトやBGは確かに便利な機能だが，その便利に甘んじていると，ハードの「固い」部分に振り回されてしまって，変わり映えのしないものしか作れない。

そもそも人間の認識機構というのは「同じものを見つけ出す」ように動機づけられている。だからたとえば，画面がすべて見慣れた16×16ドットのスプライトで埋められていれば，一応安心はする。ところがそこにひとつだけ，14×14のサイズのキャラクタが蠢いていたりすると，同じ大きさのキャラを探そうとして，ついついそれに見入ってしまうのだ。16とか8などの2の累乗数はコンピュータであれば避けることのできない数字上の制約である。しかし，ゲームの面白さを大きく左右するアイデアの，そのデキをこれまた大きく左右するうま味を引き出すためには，この制約をいかに超えるかということが大きな鍵を握っている，ということはすでに証明済みである。

話を少し戻そう。繰り返しになるが，アイデアを捻出するときには視野を広げるより，逆に思いきり狭めてしまったほうがうま味要素が加わって，アイデアの格が上がる。また論理的に見れば，厳しい条件を課して，考える範囲を思いきり狭めてしまったほうがアイデアを捻り出すにはずっと楽だ。考える範囲が狭まると目的が明確になり，全体の状況がより把握しやすくなるからである。つまり視野を狭めるという逆転の発想は，二重の意味でよいアイデアを生み出すために有利なのである。

そこでキャラグラなのだ。たとえばS-OSなら200弱の，アルファベットとか片仮名といったキャラクタが用意されているわけだが，「キャラクタしか使えない」と低く構えるのでなく，「キャラクタしか使わない」というふうに高飛車に構えてみるのである。

「そんなことをいっても，8×8の既成のキャラクタでは表現力に限界があるし，かといってその限界を超えるアイデアを捻り出す頭は自分にはない」

この期に及んでそのようなことを考える読者がいるとしたら，アナタの考えは間違っている。だいたい，そのように弱腰になること自体からして間違いの原因である。考えてみたまえ，2×2，計4個のキャラクタを使ってある形を表現するとする。すると組み合わせの総数は160億となり，そのなかから目的に適ったひとつを見つけるとなると至難の業だが，実際の状況はそれほど悲観的ではない。

走る人間を表現したいとすれば，足の部分には“)”とか“>”といった記号が使えるのではないか。また，上半身の，頭と手を1キャラで表現したいのなら“i”とか“j”などの記号を使うのがいいかもしれない。そこまで考えつけば，残りの組み合わせは数百，あるいは数十のオーダーで済む。鋭い人はとっくに気づいていると思うが，ちなみにここでも，前述の条件を設定していき選択肢を狭めていくという方法論が生きている。

実際はこのようにアイデアを積み重ねていく作業は長く辛いものである。しかし，キャラグラという極度に表現の範囲が狭められたジャンルのゲームにおいては，なにか文字が蠢いていて，目を凝らすとその形が意味を持っているように見える，という状況自体尋常でなく，同時に不思議なのであり，だからこそ人の目を引き，共感すら得られるのだ。

そういった意味で見れば，キャラグラゲームの存在の，その必然性というのはいまだに失われていない，と僕は思うのだ。

X68000における発展方向

もうだいぶ昔のことになるが，とある友人の母親と話をしていたら，ひょんなことから話題が漢字の成り立ちのことに及んだ。そこで僕は，努力の“努”の字はどうしてこういうふうに書くのか，という質問を受けた。知らないと答えると，そのおばさんはしばらく間を置いてこういったのである。

「それはね，オンナのマタにはチカラがあるからなのよ，フフフ」

そのときの，友人の母親の表情は形容しがたくミステリアスであった。それ以来おばさんの「フフフ」の表情は，僕の大脳皮質にコビリついてしまったのだ。だからたとえば，レポートなどを書いていて，努の字を書く段になると，

「オンナのマタにはチカラがある，と……，ハッ！」

などと無意識につぶやいてしまうのである。ただし最近は文章のほとんどをワープロを使って書くので，この傾向は薄れつつある。

大昔，文字というのは宗教的な儀式などといった，特別なときにしか用いられなかった。刻まれることによって，発話と同じように意味内容を伝えることができる不思議な記号は，古代人にとってはまさに魔法のようなものであったに違いない。ところが，これほど文明が発達して，文字によって伝えられる意味が，つまり文字情報の量が増えてしまうと，文字がもともと持っていた霊力みたいなものがスポイルされてしまう。

考えてみれば，キャラグラというのは表現としてはものすごくシュールだ。霊力を失ってしまった文字を，意味を超えて有機的に結びつけることによって，新しいカタチを生み出す。大袈裟かもしれないが，キャラグラには芸術に通じる部分さえあるように思う。ごくありふれたものにエネルギーを注いで，まったく違うものに昇華させるというのは，これは疑いなく芸術の仕業である。で，さっきのおばさんの「オンナのマ

タにはチカラが……」という発想も，言葉に霊力を与えうる，ものすごくシュールな発想だとはいえないか。

本当はX68000でキャラグラゲームが作れないかと思っていたのだが，ご存じのとおりX68000はビットマップ表示を採用しており，するとどこからどこまでがキャラグラだか境界線が明確でないような気がする。それならいっそキャラグラという枠を取り払ってしまって，代わりに「出てくるのはすべて漢字」という条件とスリ替えてしまえばいいのではないか。そんなゲームが作れないものだろうかと，僕は思った次第なのだ。

シュールな風景

漢字しか出てこないゲーム（多分シューティングになるだろう）というアイデアは，発想としてはありきたりだ。問題は，漢字がゲームの中でどのように使われるかという部分に集約される。

で，いつかそのゲームを作るときのためにと思って，僕はアイデアを溜めておいた。今回はそのなかの一部を，ショートゲーム風にアレンジしてお届けしよう。

リストは2つに分かれている。リスト1が外字定義用のプログラムで，リスト2はメインプログラムとなっている。両方ともBASICで書かれているので，入力の方法は説明するまでもないだろう。

ただし，リスト1の外字定義のプログラムはUSKCG.SYSを直接書き換えるようになっているので，以下の手順に従って実行していただきたい。

まず，普段使っているUSKCG.SYSのファイル名を，USKCG2.SYSなどとリネームして待避させる。そうしたらHuman68kのシステムディスクのマスターから，USKCG.SYSを普段使っているシステムディスクにコピーする。そのあとBASICを起動して，外字定義プログラムを走らせればいい。

新しいUSKCG.SYSが出来上がったら，COMMANDに戻ってUSKCGMを起動し，最初のメニューで更新を選べば外字の定義は完了である。

外字を定義してリスト2を走らせると，画面に牧歌的な風景が現れる。そこを，人が歩いていく。ヒトが歩くのではない。「人」という字が，シャナリシャナリといったふうに歩くのである。空には「因」の字が飛んでいて，また地面では，たまに「大」の字が登場する。

ここで考えてみるのである。「大」の字をよく見ると，「人」に横棒をのっけたような形をしている。そしておもむろにマウスの右ボタンをクリックすると，「大」の字がのけ反ってその横棒をひょいっと空めがけて打ち放つ。で，運よく槍が空を飛んでいる「因」に当たるとする。「因」の字もよく見ると，「大」を四角で囲った作りをしている。槍が当たれば当然囲いが取れて，とすると……。

こんなシュールなアイデアが50も60も詰め込まれたゲーム，僕の友人の母親の，「フフフ」の顔くらいインパクトがあるゲームを，いま作ろうと思っている。そのためには68000のマシン語をマスターしなければならないだろうし，またハードそのものについての知識も不十分である。来年の春くらいまでに出来上がるといいのだが。

リスト1

```
  10 /*
  20 dim char a(1008) = {
1000 /*人
1005         0, 28,  0,  0, 24,  0,  0, 24,  0,  0, 24,  0,
1010         0, 24,  0,  0, 24,  0,  0, 24,  0,  0, 56,  0,
1020         0, 56,  0,  0, 56,  0,  0, 56,  0,  0,104,  0,
1030         0,108,  0,  0,108,  0,  0,108,  0,  0,198,  0,
1040         0,198,  0,  1,135,  0,  1,131,192,  3,  1,248,
1050         2,  0,112,  4,  0, 16,  8,  0,  0, 16,  0,  0,
1060 /*が
1070         0, 28,  0,  0, 24,  0,  0, 24,  0,  0, 24,  0,
1080         0, 24,  0,  0, 24,  0,  0, 24,  0,  0, 24,  0,
1090         0, 56,  0,  0, 56,  0,  0,120,  0,  0,112,  0,
1100         0,240,  0,  0,240,  0,  0,224,  0,  0,112,  0,
1110         0,252,  0,  0,207,128,  0,131,  0,  1,128,  0,
1120         1,128,  0,  1,  0,  0,  2,  0,  0,  2,  0,  0,
1130 /*あ
1140         0, 28,  0,  0, 24,  0,  0, 24,  0,  0, 24,  0,
1150         0, 24,  0,  0, 24,  0,  0, 56,  0,  0, 56,  0,
1160         0,120,  0,  0,248,  0,  0,216,  0,  1,216,  0,
1170         1,144,  0,  1,176,  0,  0,176,  0,  0,176,  0,
1180         0,112,  0,  0,112,  0,  0, 48,  0,  0, 48,  0,
1190         0, 56,  0,  0, 16,  0,  0, 24,  0,  0, 24,  0,
1200 /*る
1210         0, 28,  0,  0, 24,  0,  0, 24,  0,  0, 24,  0,
1220         0, 24,  0,  0, 24,  0,  0, 56,  0,  0, 56,  0,
1230         0,120,  0,  0,120,  0,  0,120,  0,  0,216,  0,
1240         0,216,  0,  0,200,  0,  0,136,  0,  0,136,  0,
1250         1,140,  0,  1,140,  0,  1,  4,  0,  1,  6,  0,
1260         1,  7,  0,  1,  3,128,  1,  3,192,  0,  1,128,
1270 /*く
1280         0, 28,  0,  0, 24,  0,  0, 24,  0,  0, 24,  0,
1290         0, 24,  0,  0, 24,  0,  0, 56,  0,  0, 56,  0,
1300         0, 56,  0,  0,104,  0,  0,108,  0,  0,100,  0,
1310         0,196,  0,  0,196,  0,  0,134,  0,  1,130,  0,
1320         1,  2,  0,  3,  3,  0,  2,  1,128,  4,  1,128,
1330         8,  0,192, 16,  0,224,  0,  0,120,  0,  0, 48,
1340 /*大
1350        60,  0,  0,  3,130,  0,  0,115,128,  0, 15,  0,
1360         0,  7,128,  0, 14, 98,  0, 12, 30,  0, 28,  6,
1370         0, 24,  3,  0, 48,  1,  0, 48,  0,  0,104,  0,
1380         0,104,  0,  0, 72,  0,  0,196,  0,  0,132,  0,
1390         1,134,  0,  1,  2,  0,  2,  3,  0,  2,  1,192,
1400         4,  0,224,  8,  0,126, 48,  0, 28,192,  0, 12,
1410 /*が
1420         7,  0,  0,  0,192,  0,  0, 48,  0,  0,  8,  0,
1430         0,  6, 48,  0,  1,240,  0,  3,248,  0, 15,232,
1440         0, 31, 32,  0, 60, 16,  0,124, 16,  0,104,  8,
1450         0,216, 12,  1,144,  7,  1,144,  6,  3, 24,  2,
1460         6,  8,  0,  4, 12,  0,  8,  6,  0, 16,  7,  0,
1470        32,  3,192, 32,  1,240, 64,  0,124, 64,  0, 24,
1480 /*槍
1490         0, 24,  0,  0, 12,  0,  0,  2,  0,  0,  1,  0,
1500         0,  0,128,  0,  0, 64,  0,  0, 96,  0,  0, 32,
1510         0,  7,252,  0, 31,252,  0,126, 20,  1,216, 16,
1520         3,184,  8,  2, 48,  8,  6, 96,  8, 12, 96,  6,
1530         8, 96,  7, 24, 96,  6, 16, 32,  4, 32, 48,  0,
1540        32, 28,  0, 64, 15,192, 64,  7,128, 64,  1,128,
1550 /*を
1560         0,  0,128,  0,  0,128,  0,  0, 64,  0,  0, 64,
1570         0,  0, 32,  0,  0, 32,  0,  0, 32,  0,  0, 32,
1580         0, 28, 16,  0,127,144,  1,255,208,  3,185,240,
1590         7, 96, 56, 12,192, 28, 13,128, 28, 25,128,  8,
1600        17,128,  8, 49,128, 12, 32,192, 14, 32,224, 12,
1610        96,120,  8, 64, 63,  0, 64, 14,  0, 64,  6,  0,
1620 /*投
1630         0,  0,  0,  0,  0,  4,  0,  0,  4,  0,  0,  4,
1640         0,  0,  4,  0,  0,  4,  0,  0,  4,  0,  0,  4,
1650         0,120,  4,  1,255,  4,  7,255,132, 14,227,196,
1660        13,128,244, 27,  0,124, 22,  0, 60, 54,  0, 30,
1670        38,  0, 15, 38,  0,  6, 99,  0,  4, 67,128,  6,
1680        65,240,  7, 64,254,  6, 64, 60,  4, 64, 12,  0,
1690 /*げ
1700         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
1710         0,  0,  0,  0,  0,  0,  0,  0, 28,  0,  1,254,
1720         0, 15,254,  0, 63,194,  0,116,  0,  1,200,  0,
1730         3,144,  0,  2, 32,  0,  6, 96,  0, 12, 96,  0,
1740         8, 96,  0, 24, 96,  0, 16, 48,  0, 32, 56,  0,
1750        32, 30,  0, 64, 15,224, 64,  3,192, 64,  0,192,
1760 /*る
1770         0,  0,  0,  1,  0,  0,  3,128,  0,  7,192,  0,
1780         1,192,  0,  0,224,  0,  0, 96,  0,  0,112,  0,
1790         0,112,  0,  0, 48,  0,  0, 56,  0,  0, 56,  0,
1800         0, 52,  0,  0,100,  0,  0, 98,  0,  0, 67,  0,
1810         0,193,128,  0,128,192,  1,128,112,  1,  0, 63,
1820         2,  0, 30,  4,  0,  6, 24,  0,  0, 96,  0,  0,
1830 /*大の横棒
1840         0,  0,  0,  0,  0,  8,  0,  0, 28,127,255,254,
1850         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
1860         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
1870         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
1880         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
```

```
1890        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
1900 /*槍
1910        0,  0,  0,  0,  0,  4,  0,  0,  4,  0,  0,  4,
1920        0,  0,  4,  0,  0,  4,  0,  0,  4,  0,  0,  4,
1930        0,  0,  4,  0,  0,  4,  0,  0,  4,  0,  0,  4,
1940        0,  0,  4,  0,  0,  4,  0,  0,  4,  0,  0,  4,
1950        0,  0,  4,  0,  0,  4,  0,  0,  4,  0,  0,  6,
1960        0,  0,  7,  0,  0,  6,  0,  0,  4,  0,  0,  0 }
1970 /*
3000 int b,c,d : dim char e(71)
3010 error off
3020 b = fopen( "USKCG.SYS","w" )
3030 if b = -1 then {
3040     print"USKCG.SYS が見付かりません。"
3050     print"カレントをUSKCG.SYSのあるディレクトリに";
3060     print"移し、再実行して下さい。"
3070     end         }
3080 fseek( b , &H9FA , 0 )
3090 for c = 0 to 13
3091   for d = 0 to 71
3092       e( d ) = a( c * 72 + d )
3093   next
3100   fwrite( e , 72 , b )
3110   fseek( b , 2 , 1 )
3120 next
3130 fclose( b )
3140 end
```

リスト2

```
1000 /*##################################
1010 /*#                         シュールな風景
1020 /*#                                       14/9/'92  (ats)
1030 /*#                 (ちなみに終了は左クリック)
1040 /*##################################
1045 key 3,"screen 2,0,1,1"+chr$(13)
1050 randomize(0)
1060 int a,b,c,d
1070 dim x(4),y(4),s1(4),s2(4),k(4)
1080 dim str ch(15)
1090 for a = 160 to 174
1100  ch( a - 159 ) = chr$(235)+chr$( a )
1110 next
1120 ch( 0 ) = "人"
1130 scr() : apage( 1 ) : mouse( 4 ) : mouse( 2 )
1140  x(0) = -1 : k(0) = 1
1150  x(1) = -1 : k(1) = 1
1160  x(2) = -1 : k(2) = 3 : a = 0
1170  repeat
1180  org()
1190  a = a + 1 : if a = 512 then a = 0
1200  home( 2,a,0 )
1220   msstat(b,b,c,d)
1240 until c = -1
1250 screen 2,0,1,1 : mouse( 0 )
1260 end
1270 /*★★★★★ キャラクター制御関数  ★★★★★
1280 func org()
1290  int a,b
1300  for a = 0 to 2
1310   if k( a ) = 1 then kakoi( a )
1320   if k( a ) = 2 then hito( a )
1330   if k( a ) = 3 then dai( a )
1340  if k(a) = 0 then {
1342   for b = 0 to 2
1344    if k(b) = 3 then {
1346     k(a) =( a mod 2 ) + 1 : x(a) = -1
1348     break }
1349   next
1350   if b = 3 then {
1352     k(a) = 3 : x(a) = -1 } }
1360  next
1370   if k( 3 ) = 4 then ya( 3 )
1380  endfunc
1390 /*★★★★★★★★ 人を歩かせる  ★★★★★★★
1400 func hito( no;int )
1410  if x(no) = -1 then {
1420    x(no) = 256 : y(no) = 196
1430    s1(no) = 0  }
1440  fill( x(no),y(no),x(no)+23,y(no)+23,0 )
1450  x(no) = x(no) - 4 : s1(no) = s1(no) + 1
1460  if s1(no) = 5 then s1(no) = 0
1470  symbol( x(no),y(no),ch(s1(no)),1,1,2,10,0 )
1480  if x(no) < 0 then k(no) = 0
1490 endfunc
1500 /*★★★★★★★★★ 因を飛ばす ★★★★★★★★★
1510 func kakoi( no;int )
1520  if x(no) = -1 then {
1530     x(no) = 256 : y(no) = rnd()*50+30  }
1540  fill( x(no),y(no),x(no)+23,y(no)+23,0 )
1550  x(no) = x(no) - 10
1560  symbol( x(no),y(no),"因",1,1,2,10,0 )
1570  if x(no) < -24 then k(no) = 0
1580  if k(3) <> 0 then {
1590  if abs( y(3) - y(no) ) < 28 then {
1600   if x(3) - x(no) + 20 < 28 and x(3)+20 > x(no) then {
1610    k(no) = 3 : s2(no) = -1 } } }
1620 endfunc
1630 /*★★★★★★★★  大を歩かせる  ★★★★★★★
1640 func dai( no;int )
1650  int a,b,c
1660  if x(no) = -1 then {
1670    x(no) = 256 : y(no) = 196
1680    s1(no) = 0  :s2(no) = 0 }
1690  fill( x(no),y(no),x(no)+23,y(no)+23,0 )
1700  if s2(no) < 1 then {
1710  if s2(no) <> -1 then {
1720  x(no) = x(no) - 4 : s1(no) = s1(no) + 1
1730  } else { y(no) = y(no) + 16
1740   if y(no) > 196 then y(no) = 196 : s2(no) = 0
1750 }
1760  if s1(no) = 5 then s1(no) = 0
1770  symbol( x(no),y(no),ch(s1(no)),1,1,2,10,0 )
1780  line( x(no)+ 1,y(no)+3,x(no)+22,y(no)+3,10 )
1785  line( x(no)+20,y(no)+2,x(no)+21,y(no)+2,10 )
1790  if x(no) < 0 then k(no) = 0 : x(no) = 500
1800  if s1(3) <> 8 and k(3) = 0 and x(no) > 12 then {
1810   if d <> 0 then {
1820    s2(no) = 1 : s1(3) = 8 }
1860  } } else {
1870  fill( x(no),y(no),x(no)+23,y(no)+23,0 )
1880  symbol( x(no),y(no),ch(s2(no)+4),1,1,2,10,0 )
1890  s2(no) = s2(no) + 1
1900  if s2(no) = 7 then {
1910   k(3) = 4 : x(3) = x(no) : y(3) = 196 }
1920  if s2(no) = 8 then k(no) = 2
1930  }
1940 endfunc
1950 /*★★★★★★★★★ 槍を跳ばす ★★★★★★★★
1960 func ya( no;int )
1970  symbol( x(no),y(no),ch(13),1,1,2, 0,0 )
1980  y(no) = y(no) - s1(no)
1990  if s1(no) < 24 then s1(no) = s1(no) + 3
2000  symbol( x(no),y(no),ch(13),1,1,2,10,0 )
2010  if y(no) < -24 then k(no) = 0 : s1(no) = 0
2020 endfunc
2030 /*★★★★★★★★  画面の初期化  ★★★★★★★
2040 func scr()
2050 int a,b,c,d
2060  screen 0,1,1,1
2070 palet( 1,30672) : palet( 2,21766)
2080 palet( 3,1792 ) : palet( 4,1152 ) : palet( 5,65274 )
2090 palet( 6,62644) : palet( 7,26378) : palet( 8,57680 )
2100 palet( 9,10294) : palet(10,   0 )
2110 /*★★★★★★★★★ 舞台を描く ★★★★★★★★★
2120  apage( 0 )
2130  fill( 0,220,255,228,1 ) : fill( 0,229,255,232,2 )
2140  fill( 0,233,255,255,1 ) : fill( 0,0,255,4,1 )
2150  ci(  5,120,1024,0 ) : ci( 219,85,900,1 )
2160  ci(  5,120,1848,0 ) : ci( 5,120,5000,0 )
2170  ci( 219,85,1648,1 ) : ci( 219,85,4800,1 )
2180 fill( 0,124,5,135,3 ) : fill( 250,124,255,135,3 )
2190 /*★★★★★★★★★  雲を描く  ★★★★★★★★
2200  apage( 2 ) : window( 0,0,511,255 )
2210  for a = 1 to 9
2220   b = a*50+rnd()*10 : c = rnd()*50+20 : d = rnd()*15+20
2230   cloud( b,c,d )
2240  next
2250  cloud( 512,40,25 ) : cloud( 0,40,25 )
2260 /*★★★★★  空とお日様と木を描く  ★★★★★
2270  apage( 3 ) : window( 0,0,255,255 )
2280  fill( 0,0,255,219,9 )
2290  circle( 40,50,30,7 ) : paint( 40,50,7 )
2300  for a = 1 to 7
2310   tree()
2320  next
2330 endfunc
2340 func ci(y;int,r;int,hv;int,m;int )
2350  int s1=270,e1=360,s2=180,e2=270
2360  if m = 1 then s1=0 : e1=90 : s2=90 : e2 = 180
2370  circle(   0,y,r,4,s1,e1,hv )
2380  circle( 255,y,r,4,s2,e2,hv )
2390  paint( 0,y,3 ) : paint( 255,y,3 )
2400 endfunc
2410 func cloud( x;int,y;int,l;int )
2420  int a,b,c,d
2430  circle( x,y,l,6,180,360,128 )
2440  circle( x+l*3/4,y,l/4,6,180,360,100 )
2450  circle( x-l*1/4,y,l*3/4,5,0,180,128 )
2460  paint( x-l+2,y,5 ) : paint( x+l/2-1,y,5 )
2470 endfunc
2480 func tree()
2490  int a,b,c,d
2500  a = rnd()*25 : a=a*10 : circle( a,205,8,8,,,512 )
2510  paint( a,204,8 ) : fill( a-1,214,a+1,219,2 )
2520 endfunc
```

探しもの

Murata Toshiyuki 村田 敏幸

今回は，3つの探索アルゴリズムを紹介しましょう。線形探索，2分探索，そしてハッシュ法です。どの方法が有効かは，探索データの条件によって異なってきます。それぞれのアルゴリズムを理解して，効率よいデータ検索に挑戦してみてください。

多数のデータのなかから特定のデータを探し出す処理＝“探索/検索(searching)”はプログラムには欠くことのできない要素だ。アルゴリズム関係ネタを敬遠する読者は多いようだが，気にせずに，今回はこの探索アルゴリズムをみていきたい。

線形探索

データは配列に並んでいるものとしよう。この中に特定のデータがあるかどうか，あるとしたらその位置を知りたい。どうするか。誰でも思いつくのは配列先頭の要素から順に比較/照合していく方法だろう。この素朴な探索アルゴリズムはリニアサーチ(linear search：線形探索)と呼ばれる。

リスト1はこのリニアサーチにより32ビット整数の配列から特定の値をもつ要素を探すサブルーチンの一例だ。このサブルーチンsearchはつぎのようにして呼び出す。

```
pea.l    配列末尾アドレス
pea.l    配列先頭アドレス
move.l   検索キーデータ, -(sp)
jsr      search
lea.l    12(sp),sp
```

単純なプログラムだが，あえて丁寧に説明してみよう。3行はお馴染みの外部定義，サブルーチンsearchをほかのプログラムとリンクして使えるようにするためのお約束だ。続く5〜9行では引数構造のオフセット表を定義している。サブルーチンが呼び出された時点でスタックに積まれているはずの各引数のスタック上での位置(スタックポインタからのオフセット)を記号定数KEY, ARYST, ARYEDに定義している。サブルーチン先頭でlink命令によりフレームポインタを固定する場合は，linkによって待避されるアドレスレジスタの分を最初から考慮に入れてオフセットにもう4加えるのだが，今月はlinkを使わずに押し切る態勢だ。

.textでオフセット表の定義を終結させて(11行)，縁起ものの.evenを置いた(12行)あと，14行からサブルーチン本体が始まる。ここでは，まず，型どおりにサブルーチン内部で使用するレジスタをスタックに待避する(15行)。レジスタを待避することでスタック上の引数の相対位置がずれるので，どれだけずれるか，直後の16行で記号定数SAVSIZに定義している。データレジスタ1本，アドレスレジスタ1本，レジスタ1本あたり4バイト，と読んでほしい。プログラムを変更して，待避するレジスタの数が変わった場合に修正が最小限ですむよう，式の形で書いてある。

ちなみに，SAVSIZを定義するのにequではなく

リスト1 SEARCHO.S

```
 1: *        32ビット整数配列の線形探索
 2:
 3:          .xdef   search
 4: *
 5:          .offset 4
 6: *
 7: KEY:     .ds.l   1          *キー
 8: ARYST:   .ds.l   1          *配列先頭
 9: ARYED:   .ds.l   1          *配列末尾
10: *
11:          .text
12:          .even
13: *         .
14: search:
15:          movem.l d0/a1,-(sp)
16: SAVSIZ   =       (1+1)*4
17:          movem.l KEY+SAVSIZ(sp),d0/a0-a1
18:
19: *        d0 = キー
20: *        a0 = 配列先頭
21: *        a1 = 配列末尾
22:
23:          bra     next
24: loop:    cmp.l   (a0)+,d0           *先頭から順に比較
25:          beq     found              *見つかった
26: next:    cmpa.l  a1,a0              *末尾に達っするまで
27:          bcs     loop               *  繰り返す
28:
29:                                     *見つからなかった
30:          moveq.l #-1,d0             *Z=0
31:          bra     retn
32:
33: found:   subq.l  #4,a0              *Z=1
34: retn:    movem.l (sp)+,d0/a1
35:
36: *        Z=1 ... 一致するデータがあった
37: *                      (a0 = そのデータ)
38: *        Z=0 ... 一致するデータがなかった
39: *                      (a0 = 追加する位置)
40:
41:          rts
42:
43:          .end
```

“＝”(setの省略形)を使っているのは，SAVSIZを再定義可能にするためだ。リスト1にはひとつのサブルーチンしかないから問題はないのだが，1本のソースファイル中に複数のサブルーチンがある場合，equを使うとサブルーチンごとにSAVSIZ相当の定数に異なる名前をつけなければならなくなってしまう。ソース全体で使用する記号定数は誤って再定義してしまわないようequでしっかり定義すべきだが，ここは“＝”の甘さを利用する場面だ。

17行で引数をレジスタにとり出したら，メインループに突入する。要素数0の空の配列が渡される可能性を考慮し，ループ先頭から実行を始めずに，終了判定部に飛び込んでいる(23行)。ループの構成を変えて，終了判定部をループの頭に持ってきてもよかったのだが，ループ末尾に無条件分岐命令が余計に必要になるのを嫌った。プログラムの構造が不明確になるので強く勧めはしないが，一般に，while型の(終了判定が先頭にある)ループは，until型の(終了判定が最後にある)ループに組み替えて，終了判定部に飛び込むようにしたほうが効率はよい。Cコンパイラなんかではごくふつうにこんな最適化が行われる。

メインループの中身はリニアサーチのアルゴリズムどおりだ。配列先頭から1要素とり出しては検索キーデータと比較して(24行)，一致したらループを抜ける(25行)。この処理をポインタが配列末尾に達するまで繰り返す(26〜27行)。ポストインクリメントしている都合上，データの一致が検出された時点でポインタは一致したデータのつぎの要素を指している。その分を補正してa0が該当データを直に指すようにし(33行)，最初に待避したレジスタ内容を復帰して(34行)，サブルーチンから戻る(41行)。a0はそのままサブルーチンからの戻り値となる。

一致するデータがみつからなかった場合は24〜27行のループを抜けた時点でのa0の値(＝配列末尾のアドレス)をそのまま返す。不一致だったデータをメインルーチン側で配列末尾に追加する(かもしれない)ことを考慮しての仕様だ。なお，メインルーチン側で一致したかどうかが区別できなければ困るので，searchは一致時Z＝1，不一致時Z＝0のように探索結果をccrに反映する。30行はccrのZビットをクリアするためだけにある。

さて，こんな単純なアルゴリズム/プログラムにもかかわらず，リスト1には効率の点で細工の余地がある。ループを抜ける条件として，

・一致するデータがみつかった
・ポインタが配列末尾に達した

の2つがあることに着目しよう。このような場面では番人(sentinel)を導入するのが常套手段だ。具体的には，探索開始前に探そうとする探索キーデータ自身を番人として配列末尾に追加してしまう。すると，必ずこの位置で一致が検出されてループを抜けるから，ポインタが上限を超えて進む心配をしなくてもよくなり，終了判定部分を簡略化できる。

しかし，一般に，番人を導入するとメインルーチン側の負担が増すという問題がある。サブルーチンを呼び出すときに，配列の直後に番人を置くスペースがあることをいつも気にしなければならないのだ1)。そこで，番人は置かず，コーディング上の工夫で逃げたのがリスト2だ。アドレス比較と条件分岐の2命令によりループを構成していたのを，配列の要素数を使ったdbraによるループに置き換えた。dbraでは65,536回までのループしか組めないので，要素数がそれ以上になる場合に備えて，dbraの2重ループを使っている。たびたび登場しているように，

```
        subq.l  #1,d0
        swap.w d0
loop1： swap.w d0
loop2： ループ内容
        dbra    d0,loop2
        swap.w d0
        dbra    d0,loop1
```

により，d0.l回のループが実現できる。

ループカウンタとなる要素数は，24〜26行で配列

1) メインルーチンに負担をかけないよう，サブルーチン側で配列直後のメモリ内容を一時待避して番人を置き，あとで元に戻すという方法もある。ただし，配列の直後にメモリが実装されていない(メモリの最上位に配列が位置している)可能性を考えれば，汎用の方法とは言い難い。

リスト2 SEARCH1.S

```
 1: *        32ビット整数配列の線形探索
 2:
 3:          .xdef   search
 4: *
 5:          .offset 4
 6: *
 7: KEY:     .ds.l   1          *キー
 8: ARYST:   .ds.l   1          *配列先頭
 9: ARYED:   .ds.l   1          *配列末尾
10: *
11:          .text
12:          .even
13: *
14: search:
15:          movem.l d0-d1,-(sp)
16: SAVSIZ   =       (2+0)*4
17:          movem.l KEY+SAVSIZ(sp),d0-d1/a0
18:          exg.l   d1,a0
19:
20: *        d0 = キー
21: *        a0 = 配列先頭
22: *        d1 = 配列末尾
23:
24:          sub.l   a0,d1                *d1 = 配列サイズ
25:          bls     nfound               *要素数 ≦ 0なら不一致
26:          lsr.l   #2,d1                *d1 = 要素数
27:
28:          subq.l  #1,d1                *dbraを考慮
29:          swap.w  d1
30: loop1:   swap.w  d1
31: loop2:   cmp.l   (a0)+,d0             *先頭から順に比較
32:          dbeq    d1,loop2
33:          beq     found                *見つかった
34:          swap.w  d1
35:          dbra    d1,loop1
36:                                       *見つからなかった
37: nfound:  moveq.l #-1,d0               *Z=0
38:          bra     retn
39:
40: found:   subq.l  #4,a0                *Z=1
41: retn:    movem.l (sp)+,d0-d1
42:
43: *        Z=1 ... 一致するデータがあった
44: *                       (a0 = そのデータ)
45: *        Z=0 ... 一致するデータがなかった
46: *                       (a0 = 追加する位置)
47:
48:          rts
49:
50:          .end
```

末尾アドレスと先頭アドレスの差(＝配列の総バイト数)から逆算している。引数で要素数を渡すようにしてもよかったのだが，リスト1や以前作ったソーティングルーチンと揃える意味で，引数構成はそのまま残してある。

で，ここまでの説明どおりだと，内側のループは，

```
loop2： cmp.l   (a0)+,d0
        beq     found
        dbra    d1,loop2
```

になるが，リスト2ではこれをさらに変形して，

```
loop2： cmp.l   (a0)+,d0
        dbeq    d1,loop2
        beq     found
```

としている(31〜33行)。一致したらすぐfoundに分岐するのではなく，いったんループを抜けてから分岐するわけだ。dbraからdbeqに変わっていることに注目しよう。dbXX系の命令はもともと2つのループ終了条件判定を1命令で行うための命令だから，うまく使えば番人を置くのと同様の効果が期待できる。事実，リスト2の内側のループは，番人を置き，

```
loop:   cmp.l   (a0)+,d0
        bne     loop
```

と簡略化したループと同レベルの効率になっている。実行時間のほとんどがこのループで費やされることを考えると，サブルーチン全体の性能も番人を導入した場合と同じといってよい。

各種データへの対応

つぎの探索アルゴリズムを紹介する前に，ちょっと脱線する。リスト1，2は32ビット整数配列の検索専用だったが，これを8ビット整数や16ビット整数用に変更することを考えてみよう。

リスト1の場合，変更点は1カ所，24行だけだ。この

```
cmp.l  (a0)+,d0
```

を

```
cmp.w  (a0)+,d0
```

にすれば16ビット整数用，

```
cmp.b  (a0)+,d0
```

にすれば8ビット整数用になる。リスト2ではこれに加えて，配列の総バイト数から要素数を求める24〜26行の部分にも手を入れる。26行を，16ビット整数用にするなら，

```
lsr.l  #1,d1
```

にし，8ビット整数用にするなら削除すればよい。

条件つきアセンブルを利用すると，リスト3のようなこともできる。リスト3は1，2，4バイト整数，および，ポインタの配列用のリニアサーチルーチンを1本にまとめたものだ。アセンブル時に，

```
as /s  _BYTE lsearch
```

のように，シンボル“_BYTE”を定義すると1バイト整数配列用，以下，

“_WORD”を定義すると2バイト整数配列用

“_LONG”を定義すると4バイト整数配列用

“_PTR”を定義するとポインタの配列用

のオブジェクトが生成される。それにつれて，サブルーチン名も，

blsearch(1バイト整数配列用)

wlsearch(2バイト整数配列用)

llsearch(4バイト整数配列用)

plsearch(ポインタの配列用)

と変わる。実際に使用する際には，

```
as /s _BYTE /o blsearch lsearch
```

リスト3 LSEARCH.S

```
*         配列の線形探索
*
*         as -s_LONG -o llsearch.o lsearch
*         as -s_WORD -o wlsearch.o lsearch
*         as -s_BYTE -o blsearch.o lsearch
*         as -s_PTR  -o plsearch.o lsearch

          .include        cmpmac.h
*
          .offset 4
*
KEY:      .ds.l   1         *キー
ARYST:    .ds.l   1         *配列先頭
ARYED:    .ds.l   1         *配列末尾
CMPFNC:   .ds.l   1         *比較ルーチン(_PTR定義時のみ有効)
*
          .text
          .even
*
DEFPROC llsearch,llsearch,wlsearch,wlsearch,
          blsearch,blsearch,plsearch,plsearch
DEFPROC ullsearch,ullsearch,uwlsearch,uwlsearch,
          ublsearch,ublsearch,uplsearch,uplsearch

.ifdef  _PTR

__CMPFNC__        equ     (a1)
SAVREGS =         d0-d1/a1
SAVSIZ  =         (2+1)*4
          movem.l SAVREGS,-(sp)
          movem.l KEY+SAVSIZ(sp),d0-d1/a0-a1

.else

SAVREGS =         d0-d1
SAVSIZ  =         (2+0)*4
          movem.l SAVREGS,-(sp)
          movem.l KEY+SAVSIZ(sp),d0-d1/a0

.endif
          exg.l   d1,a0

*         d0 = キー
*         a0 = 配列先頭
*         d1 ≒ 配列末尾
*          (a1 = 比較ルーチン)

          sub.l   a0,d1                 *d1 = 配列サイズ
          bls     nfound                *要素数 ≦ 0なら不一致
          BYTE2N  d1                    *d1 = 要素数

          subq.l  #1,d1                 *dbraを考慮
          swap.w  d1
loop1:    swap.w  d1
loop2:    CMPDAT  (a0)+,d0              *先頭から順に比較
          dbeq    d1,loop2
          beq     found                 *見つかった
          swap.w  d1
          dbra    d1,loop1
                                        *見つからなかった
nfound:   moveq.l #-1,d0                *Z=0
          bra     retn

found:    subq.l  #DSIZ,a0              *Z=1

retn:     movem.l (sp)+,SAVREGS

*         Z=1 ... 一致するデータがあった
*                  (a0 = そのデータ)
*         Z=0 ... 一致するデータがなかった
*                  (a0 = 追加する位置)

          rts

          .end
```

のように“/o”スイッチでオブジェクトファイル名とサブルーチン名が一致するようにしておき，4本のオブジェクトを作成してライブラリファイルにまとめておくのがよいだろう。3〜6行のコメントをよくみてほしい。

リスト3ではとくに変わったことをしているわけではない。基本的には配列のデータ長に依存する部分に条件つきアセンブルをかませただけだ。しかし，条件つきアセンブル部分をマクロに閉じ込めることで，プログラムの見通しを損ねないよう配慮している。リスト2と見比べてもらうと，リスト2では，

```
lsr.l  #2,d1
```

だった行が，

```
BYTE2N d1
```

に，また，

```
cmp.l  (a0)+,d0
```

だった行が，

```
CMPDAT (a0)+,d0
```

に変更されているのがわかると思う。条件つきアセンブルにより，このマクロの定義を変更するのだ。

マクロを定義する部分はほかの探索ルーチンなどでも使い回しができそうだったので，なるべく汎用な形でリスト4のCMPMAC.Hに抜き出してある。AS.Xのマクロの貧弱さのおかげでだらだらと長くなってしまったが，注意深く追ってもらえば，

・データを比較するマクロCMPDAT
・データを転送するマクロMOVDAT
・要素数からバイト数を得るマクロN2BYTE
・N2BYTEの逆変換をするマクロBYTE2N
・1要素のデータ長を表す記号定数DSIZ
・DSIZが2の何乗かを表す記号定数DNSFT
・値がDSIZの倍数になるように切り捨てるためのビットマスクを表す記号定数DMASK

を条件つきアセンブルにより適切に定義しているのがみてとれるだろう。

また，リニアサーチには関係ないのだが，リスト4では，データ比較時に符号の有無を考慮するアルゴリズムに備えて，

_BGT，_BGE，_BLE，_BLT
_DBGT，_DBGE，_DBLE，_DBLT

リスト4　CMPMAC.H

```
*       探索/整列用マクロ定義

*       以下のシンボルを定義してアセンブルすると
*       配列の要素サイズ，符号の有無を変えた版が
*       生成される（ただし，_SIGNED，_LONGはダミー）
*
*       _BYTE + _SIGNED    符号付き 8ビット
*       _BYTE + _UNSIGNED  符号なし 8ビット
*       _WORD + _SIGNED    符号付き16ビット
*       _WORD + _UNSIGNED  符号なし16ビット
*       _LONG + _SIGNED    符号付き32ビット
*       _LONG + _UNSIGNED  符号なし32ビット
*       _PTR  + _SIGNED    符号付きデータへのポインタ
*       _PTR  + _UNSIGNED  符号なしデータへのポインタ

*
*       モード
*
LONGMD  equu    %0000
ULONGMD equ     %0001
WORDMD  equ     %0010
UWORDMD equ     %0011
BYTEMD  equ     %0100
UBYTEMD equ     %0101
PTRMD   equ     %1000
UPTRMD  equ     %1001

CMPMODE =       LONGMD          *デフォルト

*
*       データサイズ別の定数/マクロ
*
.ifdef  _PTR

CMPMODE =       PTRMD
DNSFT   equ     2

MOVDT__ macro   SOUR,DEST
        move.l  SOUR,DEST
        .endm

CMPDT__ macro   SOUR,DEST
        move.l  DEST,-(sp)
        move.l  SOUR,-(sp)
        jsr     __CMPFNC__
        addq.l  #8,sp
        .endm

.else   *ifndef _PTR

.ifdef  _BYTE

CMPMODE =       BYTEMD
DNSFT   equ     0

MOVDT__ macro   SOUR,DEST
        move.b  SOUR,DEST
        .endm

CMPDT__ macro   SOUR,DEST
        cmp.b   SOUR,DEST
        .endm

.else   *ifndef _BYTE

.ifdef  _WORD

CMPMODE =       WORDMD
DNSFT   equ     1

MOVDT__ macro   SOUR,DEST
        move.w  SOUR,DEST
        .endm

CMPDT__ macro   SOUR,DEST
        cmp.w   SOUR,DEST
        .endm

.else   *ifndef _WORD

CMPMODE =       LONGMD
DNSFT   equ     2

MOVDT__ macro   SOUR,DEST
        move.l  SOUR,DEST
        .endm

CMPDT__ macro   SOUR,DEST
        cmp.l   SOUR,DEST
        .endm

.endif
.endif
.endif

DSIZ    equ     1<<DNSFT                *配列要素サイズ
DMSK    equ     -DSIZ                   *DSIZの倍数に切り捨てる
                                        *  ビットマスク

*
*       符号の有無に応じた分岐マクロ
*
.ifdef  _UNSIGNED

CMPMODE =       CMPMODE|%0001

_BGT    macro   DEST
        bhi     DEST
        .endm

_BGE    macro   DEST
        bcc     DEST
        .endm

_BLE    macro   DEST
        bls     DEST
        .endm

_BLT    macro   DEST
        bcs     DEST
        .endm

_DBGT   macro   DREG,DEST
        dbhi    DREG,DEST
        .endm

```

というマクロ群も定義されている。通常，これらのマクロはただ単に頭の“_”をとったbgt，bge，ble，bltなどの条件分岐命令に置き換わる(141～171行)。が，シンボル“_UNSIGNED”が定義されると，符号を考慮しない，bhi，bcc，bls，bcsなどに展開される(107～137行)。

CMPDAT (a0),d0

のあと，d0>(a0)のときに分岐したい場合，一般には，

データが符号つきならbgt

データが無符号ならbhi

と使い分けるわけだが，代わりにマクロ_BGTを使えば，アセンブル時に符号つき数用か無符号つき数用かを切り替えられるようになる。

このほかに，リスト4ではデバッグ用，というか，アルゴリズム解析用の簡単なマクロDBGPUTCも定義してある。DBGPUTCは通常は空の命令列に置き換えられるだけで意味を持たない。シンボル“_DEBUG”が定義されているときのみ，IOCS B PUTCによる1文字出力の命令列に展開される。CMPDAT，MOVDATにはあらかじめこのDBGPUTCが埋め込んであり，シンボル“_DEBUG”を定義しておくと，比較が行われるごとに“?”，データ転送が行われるごとに“=”が画面に表示されるようになる。入力データの性質やアルゴリズムの違いにより比較回数や転送回数の変化する様子をみるのに便利だと思う。

リスト3，4では，ポインタの配列周りに気を払ってほしい。ここでいうポインタの配列とは，配列にデータ自身ではなくデータの置かれたアドレスを並べたものをいう。文字列のテーブルなんかではごくふつうにみられる形式だ。ポインタの配列から探索を行う場合，比較方法はポインタの先にあるデータの構造によって変わる。そこで，サブルーチンplsearch(シンボル“_PTR”を定義してリスト3をアセンブルしたときに生成される)では，“2つのポインタを受けとり，その先にあるデータを比較してccrに反映するサブルーチン”がどこにあるかを，第4引数で受けとるようになっている。渡されたサブルーチン先頭アドレスはa1にとり出され，比較が必

```
127: _DBGE    macro   DREG,DEST
128:          dbcc    DREG,DEST
129:          .endm
130:
131: _DBLE    macro   DREG,DEST
132:          dbls    DREG,DEST
133:          .endm
134:
135: _DBLT    macro   DREG,DEST
136:          dbcs    DREG,DEST
137:          .endm
138:
139: .else    *ifndef _UNSIGNED
140:
141: _BGT     macro   DEST
142:          bgt     DEST
143:          .endm
144:
145: _BGE     macro   DEST
146:          bge     DEST
147:          .endm
148:
149: _BLE     macro   DEST
150:          ble     DEST
151:          .endm
152:
153: _BLT     macro   DEST
154:          blt     DEST
155:          .endm
156:
157: _DBGT    macro   DREG,DEST
158:          dbgt    DREG,DEST
159:          .endm
160:
161: _DBGE    macro   DREG,DEST
162:          dbge    DREG,DEST
163:          .endm
164:
165: _DBLE    macro   DREG,DEST
166:          dble    DREG,DEST
167:          .endm
168:
169: _DBLT    macro   DREG,DEST
170:          dblt    DREG,DEST
171:          .endm
172:
173: .endif
174:
175: *
176: *        サブルーチン名を
177: *          モードに応じて定義するマクロ
178: *
179: DEFIT__  macro   NAME
180:          .xdef   NAME
181: NAME:
182:          .endm
183:
184: DEFPROC macro    NAML,NAMUL,NAMW,NAMUW,
                     NAMB,NAMUB,NAMP,NAMUP
185: .if      CMPMODE.eq.LONGMD
186:          DEFIT__ NAML
187: .elseif CMPMODE.eq.ULONGMD
188:          DEFIT__ NAMUL
189: .elseif CMPMODE.eq.WORDMD
190:          DEFIT__ NAMW
191: .elseif CMPMODE.eq.UWORDMD
192:          DEFIT__ NAMUW
193: .elseif CMPMODE.eq.BYTEMD
194:          DEFIT__ NAMB
195: .elseif CMPMODE.eq.UBYTEMD
196:          DEFIT__ NAMUB
197: .elseif CMPMODE.eq.PTRMD
198:          DEFIT__ NAMP
199: .elseif CMPMODE.eq.UPTRMD
200:          DEFIT__ NAMUP
201: .endif
202:          .endm
203:
204: *
205: *        アルゴリズム解析用マクロ
206: *
207: DBGPUTC macro    CHAR
208: .ifdef   _DEBUG
209:          movem.l d0-d1,-(sp)
210:          moveq.l #CHAR,d1
211:          moveq.l #$20,d0          *B_PUTC
212:          trap    #15
213:          movem.l (sp)+,d0-d1
214: .endif
215:          .endm
216:
217: *
218: *        データ転送マクロ
219: *
220: MOVDAT   macro   SOUR,DEST
221:          DBGPUTC '='
222:          MOVDT__ SOUR,DEST
223:          .endm
224:
225: *
226: *        データ比較マクロ
227: *
228: CMPDAT   macro   SOUR,DEST
229:          DBGPUTC '?'
230:          CMPDT__ SOUR,DEST
231:          .endm
232:
233: *
234: *        バイト数→要素数変換マクロ
235: *
236: BYTE2N   macro   DREG
237: .if      DNSFT.gt.0
238:          lsr.l   #DNSFT,DREG
239: .endif
240:          .endm
241:
242: *
243: *        要素数→バイト数変換マクロ
244: *
245: N2BYTE   macro   DREG
246: .if      DNSFT.gt.0
247:          lsl.l   #DNSFT,DREG
248: .endif
249:          .endm
```

要になったときにマクロCMPDATの中から呼び出される(リスト4：42～47行)。要するに配列と一緒に比較の専門家がついてくるので，plsearchは配列に並んだポインタの先にどんなデータがあるのか知らないままでも探索が行えてしまうというわけだ。

plsearchの動作試験用ルーチンをリスト5に用意しておいた。リスト5はキーボードから文字列を受けとっては，58～66行の文字列群から探し，一致するものがあるかどうかを表示する。空文字列の入力で実行を終える。文字列の比較ルーチンはリスト6に独立させてあるので，実行ファイルを作成するときは一緒にリンクすること。リスト5の7行，28行のstrcmpをstrcmpiに変更のうえ，リスト6の代わりにリスト7をリンクすれば，英字の大文字/小文字を区別しないで比較するようにもできる[2]。この例は，比較のルールまで，比較ルーチンの自由になることを示している。

なお，リスト4のマクロ集は探索だけではなく，整列(ソーティング)ルーチンにも対応できるようになっている。参考までに，以前作ったクイックソー

2) リスト7のstrcmpiでは，全角文字は考慮していない。

リスト5 LTEST.S

```
 1: *       lsearchの動作試験用プログラム
 2:
 3:         .include        doscall.mac
 4:         .include        const.h
 5:
 6:         .xref   plsearch
 7:         .xref   strcmp
 8: *
 9:         .text
10:         .even
11: *
12: ent:
13:         lea.l   inisp,sp
14:
15:         lea.l   linbuf,a1       *a1 = 1行入力バッファ
16:         st.b    (a1)            *最大入力文字数(255)をセット
17:         lea.l   2(a1),a2        *a2 = 正味文字列
18:
19: loop:   pea.l   (a1)            *1行入力
20:         DOS     _GETS           *
21:         pea.l   lfms(pc)        *
22:         DOS     _PRINT          *
23:         addq.l  #4+4,sp         *
24:
25:         tst.b   (a2)            *空文字列が入力されたら
26:         beq     done            *  終了
27:
28:         pea.l   strcmp          *探索
29:         pea.l   tblend(pc)      *
30:         pea.l   strtbl(pc)      *
31:         pea.l   (a2)            *
32:         jsr     plsearch        *
33:         lea.l   16(sp),sp       *
34:
35:         lea.l   fndms(pc),a0    *一致/不一致の表示
36:         beq     putms           *
37:         lea.l   nfndms(pc),a0   *
38: putms:  pea.l   (a0)            *
39:         DOS     _PRINT          *
40:         addq.l  #4,sp           *
41:
42:         bra     loop
43:
44: done:   DOS     _EXIT
45: *
46: strtbl:                         *テストデータ
47:         .dc.l   str1
48:         .dc.l   str2
49:         .dc.l   str3
50:         .dc.l   str4
51:         .dc.l   str5
52:         .dc.l   str6
53:         .dc.l   str7
54:         .dc.l   str8
55:         .dc.l   str9
56: tblend:
57: *
58: str1:   .dc.b   'move',0
59: str2:   .dc.b   'add',0
60: str3:   .dc.b   'sub',0
61: str4:   .dc.b   'muls',0
62: str5:   .dc.b   'divs',0
63: str6:   .dc.b   'bra',0
64: str7:   .dc.b   'bsr',0
65: str8:   .dc.b   'rts',0
66: str9:   .dc.b   'dbra',0
67: *
68: nfndms: .dc.b   '【不一致】',CR,LF,0
69: fndms:  .dc.b   '【一致】'
70: crlfms: .dc.b   CR
71: lfms:   .dc.b   LF,0
72: *
73:         .bss
74:         .even
75: *
76: linbuf: .ds.b   2+256   *1行入力バッファ
77: *
78:         .stack
79:         .even
80: *
81:         .ds.l   1024
82: inisp:
83:
84:         .end    ent
```

リスト6 STRCMP.S

```
*       文字列の比較

        .xdef   strcmp
*
        .offset 4
*
SOUR:   .ds.l   1       *文字列1
DEST:   .ds.l   1       *文字列2
*
        .text
        .even
*
strcmp:
        movem.l d0/a0-a1,-(sp)
SAVSIZ  =       (1+2)*4
        movem.l SOUR+SAVSIZ(sp),a0/a1
cmplp:  move.b  (a1)+,d0
        beq     eos
        cmp.b   (a0)+,d0
        beq     cmplp
cmprtn: movem.l (sp)+,d0/a0-a1

*       Z=0, C=1 ... SOUR < DEST
*       Z=1, C=0 ... SOUR = DEST
*       Z=0, C=0 ... SOUR > DEST

        rts
*
eos:    cmp.b   (a0),d0
        bra     cmprtn

        .end
```

リスト7 STRCMPI.S

```
*       文字列の比較(大文字/小文字同一視)

        .xdef   strcmpi
*
LOWER   macro   DREG    *DREGを小文字化する
        local   skip
        cmpi.b  #'A',DREG
        bcs     skip
        cmpi.b  #'Z'+1,DREG
        bcc     skip
        addi.b  #'a'-'A',DREG
skip:
        .endm
*
        .offset 4
*
SOUR:   .ds.l   1       *文字列1
DEST:   .ds.l   1       *文字列2
*
        .text
        .even
*
strcmpi:
        movem.l d0-d1/a0-a1,-(sp)
SAVSIZ  =       (2+2)*4
        movem.l SOUR+SAVSIZ(sp),a0/a1
cmplp:  move.b  (a1)+,d1
        beq     eos
        LOWER   d1
        move.b  (a0)+,d0
        LOWER   d0
        cmp.b   d0,d1
        beq     cmplp
cmprtn: movem.l (sp)+,d0-d1/a0-a1

*       Z=0, C=1 ... SOUR < DEST
*       Z=1, C=0 ... SOUR = DEST
*       Z=0, C=0 ... SOUR > DEST

        rts
*
eos:    move.b  (a0)+,d0
        LOWER   d0
        cmp.b   d0,d1
        bra     cmprtn

        .end
```

トルーチンをCMPMAC.Hを使って各種データ型に対応されたものをリスト8に示そう。『グラフィックス編』単行本の添付ディスクにも似たようなプログラムを収めたのだが，あれは32ビット整数と16ビット整数にしか対応していなかった(しかも，1カ所バグっていた)のに対して，リスト8はより多くのデータ型に対応している。

2分探索

配列上のデータがn個ある場合，リニアサーチでは平均n/2回，最悪n回の比較操作を必要とする。言い換えると，リニアサーチの処理時間はnに比例する。これに対して，リニアサーチより数段高速な探索アルゴリズムとして知られるバイナリサーチ(binary search：2分探索)では探索にnの対数に比例する時間しか必要としない。比較回数は最悪でも$\log_2 n$回程度ですむ。つまり，データ数nが2倍になっても，比較回数は1回しか増えない。リニアサーチではnが2倍になれば比較回数も2倍になることを考えると，これはかなりの性能向上を意味する。

ただし，バイナリサーチでは配列があらかじめソートされていることを前提にする。その点，リニアサーチよりも初期コストはかかる。また，比較1回当たりの処理時間もやや長めだ。実装方法にもよるが，データ数が10数～数10個ぐらいまではリニアサーチのほうがかえって速い。

バイナリサーチでは，まず，配列の中央の要素mと探索キーデータxを比較する。一致すればよし。一致しなければ，xとmの大小関係よりxが配列中央より手前にあるか，後ろにあるかを判断する。仮に配列が昇順(小さい順)にソートされているとすると，x＜mなら中央より手前，x＞mなら中央より後ろにしかxは存在しえない。すかさず配列を中央で分割して，xが存在する可能性のある半区間に注目し，反対側は捨てる。つぎにxと比較するのは，残った半区間の中央の要素だ。2度目の比較で探索範囲はまた半分になる。以下同様に，中央の要素と比較しては区間を半分に切り詰めていき，区間がこれ以上分割できなくなった時点で，xに一致するデータがなかったことがわかる。

念のため，1～9のデータが並んでいる中から，

リスト8 QSORT.S

```
*        配列を昇順にソートする
*
*        as -s_SIGNED   -s_LONG -o  lsort.o qsort
*        as -s_UNSIGNED -s_LONG -o ulsort.o qsort
*        as -s_SIGNED   -s_WORD -o  wsort.o qsort
*        as -s_UNSIGNED -s_WORD -o uwsort.o qsort
*        as -s_SIGNED   -s_BYTE -o  bsort.o qsort
*        as -s_UNSIGNED -s_BYTE -o ubsort.o qsort
*        as -s_SIGNED   -s_PTR  -o  psort.o qsort
*        as -s_UNSIGNED -s_PTR  -o upsort.o qsort

        .include        cmpmac.h
*
M       equ     15              *区間中の要素数がM未満になったら
                                *単純挿入法に切り替える
*
        .offset 4               *引数構造
*
ARYST:  .ds.l   1               *配列先頭
ARYED:  .ds.l   1               *配列末尾
CMPFNC: .ds.l   1               *比較ルーチン（_PTR定義時のみ有効）
*
        .text
        .even
*
DEFPROC lsort,ulsort,wsort,uwsort,
        bsort,ubsort,psort,upsort

qsort:
.ifdef  _PTR

__CMPFNC__      equ     (a5)
SAVREGS =       d0-d3/a0-a5
SAVSIZ  =       (4+6)*4
        movem.l SAVREGS,-(sp)
        movem.l ARYST+SAVSIZ(sp),a0/a3/a5

.else

SAVREGS =       d0-d3/a0-a4
SAVSIZ  =       (4+5)*4
        movem.l SAVREGS,-(sp)
        movem.l ARYST+SAVSIZ(sp),a0/a3

.endif
                                *a0 = 配列先頭
                                *a3 = 配列末尾+1
        moveq.l #DMSK,d2        *d2 = 下位ビットマスクデータ
        moveq.l #M*DSIZ,d3      *d3 = 切り替え点*DSIZ
        clr.l   -(sp)           *スタックの底マーク

        bra     loop0

rerty:  movea.l d0,a0           *a0 = 区間先頭
        movea.l (sp)+,a3        *a3 = 区間末尾+1

loop0:  move.l  a3,d0           *
        sub.l   a0,d0           *d0 = 区間バイト数
        cmp.l   d3,d0           *残り区間が十分短くなったら
        bcs     isort           *  単純挿入法に切り替える

        lsr.l   #1,d0           *d0 = 区間バイト数/2
        and.l   d2,d0           *d0 = 中央のインデックス
        beq     next0           *要素数が1以下なら分割完了

        lea.l   0(a0,d0.l),a4   *a4 = 区間中央
        MOVDAT  (a4),d0         *d0 = 境界値
        movea.l a0,a1           *a1 = 左端ポインタ
        movea.l a3,a2           *a2 = 右端ポインタ

        bra     next1
                        *分割する
loop1:  MOVDAT  -(a1),d1        *2要素を交換する
        MOVDAT  (a2),(a1)+      *
        MOVDAT  d1,(a2)         *
next1:
loop2:  CMPDAT  (a1)+,d0        *左側から位置の狂った
        _BGT    loop2           *  要素を探す
loop3:  CMPDAT  -(a2),d0        *右側から位置の狂った
        _BLT    loop3           *  要素を探す

        cmpa.l  a1,a2           *ポインタがすれ違うまで
        bcc     loop1           *  繰り返す

        subq.l  #DSIZ,a1        *行きすぎた分を補正する
                        *分割完了
        cmpa.l  a4,a1
        bcc     right

left:   movem.l a1/a3,-(sp)     *右区間をスタックに積んでおき
        movea.l a1,a3           *左区間の分割を
        bra     loop0           *  先に行う

right:  movem.l a0/a1,-(sp)     *左区間をスタックに積んでおき
        movea.l a1,a0           *右区間の分割を
        bra     loop0           *  先に行う
*
isort:                          *単純挿入法
        lea.l   -DSIZ(a3),a2    *
        bra     inext           *
iloop:  MOVDAT  (a1)+,d1        *
        CMPDAT  d1,d0           *
        _BLE    found           *
        MOVDAT  d1,-DSIZ*2(a1)  *
        cmpa.l  a3,a1           *
        bcs     iloop           *   『グラ編』添付ディスクでは
        addq.l  #DSIZ,a1        *←  この行がバグってます
found:  MOVDAT  d0,-DSIZ*2(a1)  *
inext:  movea.l a2,a1           *
        MOVDAT  -(a2),d0        *
        cmpa.l  a0,a2           *
        bcc     iloop           *
*
next0:  move.l  (sp)+,d0
        bne     rerty

done:   movem.l (sp)+,SAVREGS
        rts

        .end
```

６を探す場合を例にアルゴリズムの流れを追ってみよう。

1)　中央位置の要素である５と比較する

１２３４５６７８９

2)　５＜６だから，求める６は配列後半にある。５以下の前半部を切り捨て，新区間(ほぼ)中央位置の８と比較する

６７８９

リスト9　BSEARCH.S

```
 1: *        配列の２分探索
 2: *
 3: *        as -s_SIGNED   -s_LONG -o  lbsearch.o bsearch
 4: *        as -s_UNSIGNED -s_LONG -o ulbsearch.o bsearch
 5: *        as -s_SIGNED   -s_WORD -o  wbsearch.o bsearch
 6: *        as -s_UNSIGNED -s_WORD -o uwbsearch.o bsearch
 7: *        as -s_SIGNED   -s_BYTE -o  bbsearch.o bsearch
 8: *        as -s_UNSIGNED -s_BYTE -o ubbsearch.o bsearch
 9: *        as -s_SIGNED   -s_PTR  -o  pbsearch.o bsearch
10: *        as -s_UNSIGNED -s_PTR  -o upbsearch.o bsearch
11:
12:          .include       cmpmac.h
13: *
14:          .offset 4
15: *
16: KEY:     .ds.l   1      *キー
17: ARYST:   .ds.l   1      *配列先頭
18: ARYED:   .ds.l   1      *配列末尾
19: CMPFNC:  .ds.l   1      *比較ルーチン
20: *
21:          .text
22:          .even
23: *
24: DEFPROC lbsearch,ulbsearch,wbsearch,uwbsearch,
         bbsearch,ubbsearch,pbsearch,upbsearch
25:
26: .ifdef  _PTR
27:
28: __CMPFNC__      equu    (a3)
29: SAVREGS =       d0-d2/a1-a3
30: SAVSIZ  =       (3+3)*4
31:         movem.l SAVREGS,-(sp)
32:         movem.l KEY+SAVSIZ(sp),d0/a1-a3
33:
34: .else
35:
36: SAVREGS =       d0-d2/a1-a2
37: SAVSIZ  =       (3+2)*4
38:         movem.l SAVREGS,-(sp)
39:         movem.l KEY+SAVSIZ(sp),d0/a1-a2
40:
41: .endif
42:
43: *       d0 = キー
44: *       a1 = 配列先頭
45: *       a2 = 配列末尾
46: *        (a3 = 比較ルーチン)
47:
48:         moveq.l #DMSK,d2        *d2 = 要素サイズの倍数に
49:                                 *       切り詰めるマスク
50:
51:         movea.l a2,a0           *要素数 ≦ 0 なら終了
52:         cmpa.l  a2,a1           *
53:         bcc     nfound          *
54:
55: loop:   move.l  a2,d1           *d1 = 区間バイト数
56:         sub.l   a1,d1           *
57:         lsr.l   #1,d1           *d1 = 区間中央位置までの
58:         and.b   d2,d1           *      先頭からのバイト数
59:         lea.l   0(a1,d1.l),a0   *a0 = 区間中央位置
60:
61:         CMPDAT  (a0),d0         *中央位置の要素と比較
62:         _BGT    higher          *あるとしたら中央より後ろ
63:         beq     retn            *一致したら終了(Z=1)
64:
65:                       *中央より前
66: lower:  movea.l a0,a2           *a2 = 新区間末尾
67:         cmpa.l  a2,a1           *先頭 < 末尾のあいだ
68:         bcs     loop            *  繰り返す
69:         bra     nfound          *見つからなかった
70:                                 *(a0 = 挿入位置)
71:
72:                       *中央より後ろ
73: higher: lea.l   DSIZ(a0),a1     *a1 = 新区間先頭
74:         cmpa.l  a2,a1           *先頭 < 末尾のあいだ
75:         bcs     loop            *  繰り返す
76:
77:         movea.l a1,a0           *見つからなかった
78:                                 *(a0 = 挿入位置)
79:
80: nfound: moveq.l #-1,d0          *Z=0, N=1
81: retn:   movem.l (sp)+,SAVREGS
82:
83: *       Z=1 ... 一致するデータがあった
84: *                (a0 = そのデータ)
85: *       Z=0 ... 一致するデータがなかった
86: *                (a0 = 挿入位置)
87:
88:         rts
89:
90:         .end
```

3)　８＞６。８以上を切り捨て，新区間中央位置の７と比較する

６７

4)　７＞６。７以上を切り捨て，新区間中央位置の６と比較する。一致するデータがみつかる。

６

では，実装例をリスト９に示す。CMPMAC.Hを使っているので，データ長，および，符号の有無はアセンブル時に指定できる。具体的な指定方法は３～10行のコメントに書いてある。動作試験にはリスト５を適当に修正のうえ，流用してほしい。その際には，検索対象データがソートされていなければならないことに注意して，手作業で並べ替えるなり，プログラム冒頭でソートルーチンを使ってソートすること。

55行からがメインループだ。頭の55～59行で区間の中央位置をa0に得る。a1に区間先頭，a2に区間末尾が入っているから，その差から区間のバイト数を求め(55～56行)，２で割り(57行)，結果が配列のデータ長の倍数になるように下位ビットを適切にマスクして(58行)，改めて区間先頭アドレスに足している(59行)。そういえば，同じような処理はクイックソートルーチンにも出てきた。

区間中央を求めたら，その位置のデータｍと検索キーデータｘを比較して(61行)，

ｘ＜ｍ

ｘ＝ｍ

ｘ＞ｍ

に場合分けする(62～63行)。ｘ＞ｍのケースをはじき，ｘ＝ｍのケースをはじき，どちらでもなければｘ＜ｍの場合の処理に流れ込んでいる。２つの分岐命令の順序には注意したい。逆にしてもプログラムの動作には支障はないが，ｘ＝ｍになる確率は比較的低いことを計算に入れ，平均の処理速度が速くなるほうを選んである。

ｘ＜ｍだった場合は，66行で区間の前半部を新区間に設定する。ｘ＜ｍだった場合の73行では逆に区間の後半部を新区間とする。この段階で，新区間中にデータがまだあるようならループして，処理を繰り返す。１個もデータがなかった場合は探索失敗で戻る。リニアサーチのときと同様，探索に失敗したときは，みつからなかったデータが挿入されるべき位置をサブルーチンからの戻り値とする。挿入位置は最後まで残っていたデータの直前か，直後ということになるが，どちらになるかは最後に行った比較結果から判断できる。最後の比較結果がｘ＜ｍなら直前，ｘ＞ｍなら直後となるのはいうまでもない。

バイナリサーチはアルゴリズムがシンプルで簡単に実現でき，かつ，実用上十分な探索速度を備える

よいアルゴリズムだ。たいていの用途に第1候補として勧めることができる。しかし，純粋に探索だけを行うのではなく，平行してデータの登録を行うときには少々問題もある。配列に対するデータの挿入時には以降のデータをごっそりずらすことになるが，このデータ転送には総データ数nに比例した時間が必要だ(平均n/2個が移動する)。探索がいくら速くても，挿入にかかる時間が足を引っ張る。挿入を頻繁に行うようなら，データを配列ではなく2分木(binary tree)でもつような変形を検討するべきだ。このバイナリサーチの亜種はツリーサーチ(treesearch：木探索)と呼ばれる。が，ツリーサーチについては，“木構造”をテーマにした回かなんかで改めてとり上げることにし，今回は読者の自由課題としたい。

ハッシュ法

ハッシュ法(hash method)は，文字列など，データひとつあたりが比較的長い場合に特に有効な探索アルゴリズムだ。アセンブラやコンパイラなどの識別子の管理などによく利用される。

ハッシュ法ではまずデータを一定の方法でハッシュ値と呼ばれる小さな整数に写像する。ハッシュ値の求め方(ハッシュ関数という)には特にこうしなければならないという決まりはない。同一のデータがいつも同じハッシュ値に変換されることが保証されていれば，どんなハッシュ関数を使おうとアルゴリズムは働く(あとで触れるように，効率には影響する)。

ハッシュ値は，ある意味で“データのチェックサムのようなもの”だ。2つのデータのハッシュ値が異なれば，当然，データどうしも一致しない。このことを利用すると，探索対象データをかなり絞り込めることは容易に想像できる。

実際のハッシュ法では，ハッシュ値をいちいち比較することさえしない。検索対象データ群はハッシュ値を添え字とする配列(ハッシュ表)で管理される。ハッシュ値が0のデータはハッシュ表の第0エントリに，ハッシュ値が1のデータは第1エントリに，という具合に探索対象データを登録しておくのだ。こうしておけば，探索キーデータと同じハッシュ値をもつデータがすぐにとり出せる。特に，ハッシュ表上の該当エントリが空だった場合は，その時点で探索が不成功だということがわかる。

ただ，複数のデータが同一のハッシュ値をとる場合があるので，話はもう少し複雑だ。ハッシュ値が衝突した場合，つまり，データをハッシュ表に登録しようとしたときに，すでにそこが埋まっていた場合の対応策を検討しておかなければならない。衝突時の処理方法には大きく分けて2種類がある。ひとつは，空いている別の場所をみつけてそこに登録する方法[3]。この方法は比較的実現が容易ながら，扱えるデータの総数がハッシュ表の大きさに制限されるという欠点を持つ。また，別の格納場所の探し方によっては，ハッシュ表の一部にデータが固まってしまって探索効率を落とすことも多い。もうひとつの方法は，同一のハッシュ値をとるデータを線形リストの形で保持し，ハッシュ表にはその最初のデータへのポインタを入れておく方法だ[4]。こちらには，そう目立った欠点はなく，あえていうなら線形リストを使用するために若干メモリ消費量が増えるという程度。使えるメモリが極端に少ない場合を除けば，気にするほどのことではあるまい。今回は，こちらを採用することにしよう。

3) 開放番地方式(open addressing)という。

4) 直接連鎖(direct chaining)という。

さて，ハッシュ法による探索の効率は，異なるデータが同一のハッシュ値をとる確率に左右される。いっさいの衝突が発生しなければ最高でも1回の比較で探索が行えるが，衝突が多くなるとそれだけ比較回数も増える。このためハッシュ法の実現にあたっては，ハッシュ値の衝突回数をいかに減らすかに注意が払われる。明らかに，ハッシュ表を大きくすれば衝突の確率は下がる。加えて，なるべく均等に

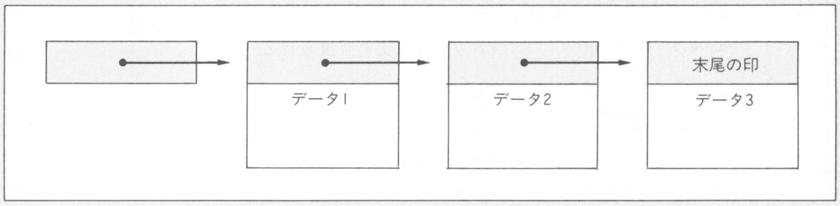
線形リスト

線形リストとはデータがポインタで1次元的に連結されたデータ構造だ。線形リスト上の各データは正味データに加えて，つぎのデータがどこにあるかをポインタでもつ(図)。末尾の1データだけは“つぎのデータがない”ことを特別な値のポインタ(通常0)で示す。

線形リストは1次元的なデータ構造という点では配列に似ている面もあるが，配列が基本的には最初に確保されたメモリの大きさに縛られるのに対して，線形リストはポインタでデータを繋いでやることで，いくらでも伸ばすことができる。また，ポインタを繋ぎ替えることで，リスト上の要素の位置を交換したり，途中にデータを挿入したり，途中のデータを削除したりといったことも配列よりずっと簡単に行える。

逆に，線形リストの欠点としては，いわゆるランダムアクセスが不可能な点が挙げられる。データを参照するときには先頭から順にポインタを辿っていかなければならず，配列のようにいきなり途中の要素をとり出すといったことができない。さらに，コストの高さも不利な点だ。ポインタを保持する分だけ配列よりも余計なメモリが必要であり，また，ポインタで間接参照するために速度効率も多少落ちる。

図　線形リスト



ハッシュ表を使うようなハッシュ関数を選ぶことが重要だ。もっとも，どんなデータ群に対しても，常に一様なばらつきを持ったハッシュ値を返すハッシュ関数というものは存在しえない。ある程度，データ群の傾向を予測して，平均的なデータに対してうまく働くよう，試行錯誤することも必要になってくる[5]。

ところで，ここまでの説明はハッシュ値の範囲とハッシュ表の大きさが揃っていることが前提になっている。この前提を満たすためには，比較的大きめの値で求めた仮のハッシュ値をハッシュ表の大きさで割り，その余りをハッシュ値として使うのが自然だ。ここで，ハッシュ表の大きさを素数にとると，余りを求める演算自体にある程度ハッシュ値をばらまく効果が期待できる。逆に，ビットマスクを使って剰余を求める手間を軽減しようとハッシュ表の大きさを2^nの値なんかにすると，データ自体の傾向やハッシュ関数の癖が強調されて，ばらつきが悪くなることもある。

では，リスト10にハッシュ法による探索ルーチンhsearchを示す。ハッシュ関数はhsearchには組み込んでしまわずに，引数として計算ルーチンの先頭アドレスを渡すようにした。この計算ルーチンはハッシュ値ではなくハッシュ表上の位置をa0に直接返すものとする。こういう仕様だから，hsearchはハッシュ値の求め方はもちろん，ハッシュ表がどこにあるかも知らない。

リスト10は，リスト3や9とは異なり，探索が不成功に終わった場合はデータを登録してから戻るようになっている。また，主に文字列の探索に使用することを想定して，探索キーデータは常にポインタで受けとる。CMPMAC.Hをとり込んではいるが，DBGPUTCを使っているだけなので，“_DEBUG”以外のシンボルを定義しても何も起こらない。

プログラムは大きく3つの部分に分けられる。39～40行でハッシュ表上の位置を求め，42～51行で該当位置に線形リストとして繋がれたデータと順に比較し，56～80行で不一致だった場合のハッシュ表への登録を行っている。このうち，ややごちゃついているのが，最後のハッシュ表への登録部分だ。この部分の繁雑さは，動的にメモリを確保している点にある。線形リストにデータを繋ぐ際にはデータ自身とデータ間の繋がりを表すポインタを格納できるだけのメモリが必要だ。しかし，どれだけのメモリを確保すればよいかは事前には予測できないのがふつうなので，必要になったときにどこからか調達する必要がある。リスト10ではこのメモリ調達をmallocというサブルーチンに一任することにした。malloc自身はこのリストには含まれていないが，とにかくmallocに何バイトよこせといえば，どこからかメモリを切り出してきてくれるものとする。ところが，hsearch自身は自分が扱っているデータについては

5）もし，探索対象が固定であれば，全データのハッシュ値が重複しないところまで，ハッシュ関数を煮詰めることも可能だ。このようなハッシュ関数を完全ハッシュ関数といい，特に，n個のデータを0～n－1に変換する場合を最小完全ハッシュ関数という。

リスト10 HSEARCH.S

```
 1: *        ハッシュ法による探索
 2: *
 3: *        as hsearch
 4:
 5:          .include        cmpmac.h
 6: *
 7:          .xdef   hsearch
 8:          .xdef   uhsearch
 9:          .xref   malloc
10: *
11:          .offset 0         *節の構造
12: *
13: NEXT:    .ds.l   1         *リンクポインタ
14: DATA:                      *正味データ（不定長）
15: *
16:          .text
17: *
18:          .offset 4
19: *
20: KEY:     .ds.l   1         *キー
21: HSHFNC:  .ds.l   1         *ハッシュ表位置計算ルーチン
22: CMPFNC:  .ds.l   1         *比較ルーチン
23: SIZFNC:  .ds.l   1         *データサイズ取得ルーチン
24: *
25:          .text
26:          .even
27: *
28: hsearch:
29: uhsearch:
30:          movem.l d0/a1-a3,-(sp)
31: SAVSIZ   =       (1+3)*4
32:          movem.l KEY+SAVSIZ(sp),d0/a1-a3
33:
34: *        d0 = キー
35: *        a1 = ハッシュ表位置計算ルーチン
36: *        a2 = 比較ルーチン
37: *        a3 = データサイズ取得ルーチン
38:
39:          move.l  d0,-(sp)           *ハッシュ表上の位置を得る
40:          jsr     (a1)               *
41:
42: loop:    move.l  (a0),d0            *d0 = つぎの節
43:          beq     nfound
44:          movea.l d0,a0              *a0 = 注目している節
45:
46:          DBGPUTC '?'                *デバッグ用
47:
48:          pea.l   DATA(a0)           *比較する
49:          jsr     (a2)               *
50:          addq.l  #4,sp              *
51:          bne     loop               *一致しなければさらに辿る
52:
53:          bra     retn               *見つかった(Z=1)
54:
55:                             *見つからなかったので新規登録
56: nfound:  jsr     (a3)               *キーデータのバイト数を取得
57:          addq.l  #DATA,d0           *リンクポインタの分
58:          movea.l a0,a1              *a1 = リンクする位置
59:
60:          move.l  d0,-(sp)           *必要なだけメモリを確保
61:          jsr     malloc             *
62:          addq.l  #4,sp              *
63:          bmi     retn               *メモリ不足だった(N=1)
64:
65:          move.l  a0,(a1)            *リンクする
66:
67:          movea.l (sp),a1            *a1 = 登録するデータ
68:          movea.l a0,a2              *a2 = いま作成した節
69:
70:          clr.l   (a2)+              *チェインリンク末尾の印
71:
72:          subq.l  #1,d0              *dbraを考慮
73:          bcc     cpylp2
74:          bra     done
75:
76: cpylp1:  swap.w  d0                 *登録するデータを
77: cpylp2:  move.b  (a1)+,(a2)+        *  いま確保したメモリに
78:          dbra    d0,cpylp2          *  コピーする
79:          swap.w  d0                 *
80:          dbra    d0,cpylp1          *
81:
82: done:    moveq.l #1,d0              *Z=0, N=0
83: retn:    addq.l  #4,sp
84:          movem.l (sp)+,d0/a1-a3
85:
86: *        Z=1, N=0 ... 一致するデータがあった
87: *                     (a0 = そのデータ)
88: *        Z=0, N=0 ... 新規に登録した
89: *                     (a0 = そのデータ)
90: *             N=1 ... 一致するデータがなく
91: *                         メモリ不足のため
92: *                         登録もできなかった
93:
94:          rts
95:
96:          .end
```

何も知らないので，何バイト要求したらいいのか，それすらもわからない。知っているのは，hsearchを呼び出したメインルーチンだけだ。そこで，hsearchでは“データの長さを返すサブルーチン”のアドレスを引数としてあらかじめ受けとるようになっている。

これだけの情報を示せば，56〜80行で何をやっているかはだいたいわかるだろう。追加するデータのバイト数を得(56行)，ポインタを格納する分の4バイトを加えて必要なメモリ量を求める(57行)。求めたバイト数をmallocに渡すと，mallocは指定サイズのメモリブロックをa0に返す(60〜63行)。このメモリブロックをa1の指す線形リスト末尾に繋げ(65行)，線形リスト末尾であるという印をつけて(70行)，正味データをコピーすれば(72〜80行)，ハッシュ表への登録は完了する。

残るmallocの実現だが，今回はまともなメモリ管理ルーチンを示すゆとりがない。そこで，これ以上簡単にはならない，超弩級の手抜き版メモリ管理ルーチンをリスト11に用意してお茶を濁す。リスト11のmallocはあらかじめinitheapで確保/初期化された大きなメモリブロック(仮にヒープと呼ぶ)の先頭から少しずつ切り出しては返すだけのサブルーチンだ。ヒープの使用中部分と未使用部分はポインタcurptrの指す位置で分割され，mallocは呼び出しごとにcurptrを戻り値とし，同時に確保した分だけcurptrを進める。Cのmalloc関数で確保したメモリはfree関数でヒープに返却できるものだが，リスト11では一度確保されたメモリがヒープに戻されることはない。freeというサブルーチンも定義してはあるが，ダミーだ。

では，hsearchの動作試験用プログラムを示して終わろう。リスト12だ。このプログラムは，キーボードから入力した文字列が未知のものだったらハッシュ表に登録し，すでに表にあればその旨表示する。改行のみの入力で終了し，最後にハッシュ表の使用効率を表示する(108〜111行のコメント参照)。

リスト12では，ハッシュ関数として，文字列の各文字の文字コードを3ビット左ローテートしてはXORで重ねて16ビットの値を作り，これをハッシュ表の大きさで割って余りをとる，といういい加減なものを使った(67〜78行)。テストに使ったサンプルデータ[6]ではそれなりのばらつきを示していたようだが，これが最良というわけではない。読者もいろいろなハッシュ関数を作って試してみてもらいたいと思う。なお，ハッシュ表の大きさは6行で仮に1021に定義してあるが，動作試験時はもっと小さな値にしたほうがハッシュ関数の性格が出やすいだろう。また，ハッシュ表の大きさが素数の場合とそうでない場合の比較もしておいてほしい。

＊

というわけで，今回は比較的ポピュラーな探索アルゴリズムを紹介してみた。次回は，いまさらながらHuman68k Ver.2.0の機能について触れてみたい(たぶん，前後編だな)。

6) サンプルデータとしては，手元にあった英文ドキュメントの冒頭から10000語を選び，重複をとり除いたあと残った1400語ほどを使った。

リスト11 TINYHEAP.S

```
*       簡易メモリ管理ルーチン

        .xdef   initheap
        .xdef   malloc
        .xdef   free
*
        .offset 4       *initheap
*
HEAPST: .ds.l   1       *ヒープ先頭
HEAPED: .ds.l   1       *ヒープ末尾
*
        .offset 4       *malloc
*
SIZE:   .ds.l   1       *確保するメモリ量
*
        .offset 4       *free
*
MEMPTR: .ds.l   1       *解放するメモリブロック
*
        .text
        .even
*
initheap:
        move.l  HEAPST(sp),heapst
        move.l  HEAPST(sp),curptr
        move.l  HEAPED(sp),heaped
        rts
*
malloc:
        move.l  d0,-(sp)
SAVSIZ  =       (1+0)*4
        move.l  SIZE+SAVSIZ(sp),d0
        movea.l curptr(pc),a0
        add.l   a0,d0
        cmp.l   heaped(pc),d0
        bls     okretn
        moveq.l #-1,d0          *N=1
        bra     retn
okretn: addq.l  #1,d0
        andi.b  #$fe,d0
        move.l  d0,curptr
        moveq.l #0,d0           *N=0
retn:   movem.l (sp)+,d0        *ccr不変
        rts
*
free:                   *ダミー
        rts
*
heapst: .ds.l   1       *ヒープ先頭
heaped: .ds.l   1       *ヒープ末尾
curptr: .ds.l   1       *次に割り当てるメモリ

        .end
```

リスト12 HTEST.S

```
 1: *       hsearchの動作試験用プログラム
 2:
 3:         .include        doscall.mac
 4:         .include        const.h
 5: *
 6: HASHSIZ equ     1021            *ハッシュテーブルの大きさ
 7: *
 8:         .xref   hsearch
 9:         .xref   strcmp
10:         .xref   initheap
11: *
12:         .text
13:         .even
14: *
15: ent:
16:         lea.l   inisp,sp
17:
18:         pea.l   heaped          *ヒープを初期化する
19:         pea.l   heapst          *
20:         jsr     initheap        *
21:         addq.l  #8,sp           *
22:
23:         lea.l   linbuf,a1       *a1 = 1行入力バッファ
24:         st.b    (a1)            *最大入力文字数(255)をセット
25:         lea.l   2(a1),a2        *a2 = 正味文字列
26:
27: loop:   pea.l   (a1)            *1行入力
28:         DOS     _GETS           *
29:         pea.l   lfms(pc)        *
30:         DOS     _PRINT          *
```

```
        addq.l  #4+4,sp         *

        tst.b   (a2)            *空文字列が入力されたら
        beq     done            *  終了

        pea.l   strlenz(pc)     *探索
        pea.l   strcmp          *
        pea.l   hash(pc)        *
        pea.l   (a2)            *
        jsr     hsearch         *
        lea.l   16(sp),sp       *

        bmi     nomem           *メモリ不足

        lea.l   fndms(pc),a0    *一致/不一致の表示
        beq     putms           *
        lea.l   nfndms(pc),a0   *
putms:  pea.l   (a0)            *
        DOS     _PRINT          *
        addq.l  #4,sp           *

        bra     loop

done:   bsr     hashreport

        DOS     _EXIT

*
*       ハッシュ表上の位置を得る
*
hash:
DPTR    =       4
        movem.l d0-d1,-(sp)
SAVSIZ  =       (2+0)*4
        movea.l DPTR+SAVSIZ(sp),a0

        moveq.l #0,d0           *ハッシュ値を求める
hloop:  move.b  (a0)+,d1        *
        beq     hdone           *
                                *
        rol.w   #3,d0           *
        eor.b   d1,d0           *
        bra     hloop           *

hdone:  divu.w  #HASHSIZ,d0     *ハッシュ表の大きさに
                                *  収める
        clr.w   d0              *
        swap.w  d0              *

        lea.l   hshtbl,a0       *a0 = ハッシュ値に対応する
        lsl.l   #2,d0           *      表上の位置
        adda.l  d0,a0           *

        movem.l (sp)+,d0-d1
        rts

*
*       文字列の長さを得る
*       (末尾の0の分を含む)
*
strlenz:
STR     =       4
        move.l  a0,-(sp)
SAVSIZ  =       (0+1)*4
        movea.l STR+SAVSIZ(sp),a0

        moveq.l #0,d0
lenlp:  addq.l  #1,d0
        tst.b   (a0)+
        bne     lenlp

        movea.l (sp)+,a0
        rts

*
*       ハッシュ表の使用効率を表示する
*
*               SPACE ... 空きエントリ
*               .     ... 衝突なし
*               1-9   ... 衝突回数
*               +     ... 衝突回数10回以上
*
hashreport:
        lea.l   hshtbl,a0
        movea.l a0,a1
        pea.l   (a1)

        move.w  #HASHSIZ-1,d7
ploop1: moveq.l #' ',d1
        move.l  (a0)+,d0
        beq     putc

        moveq.l #'.',d1
        movea.l d0,a2
        move.l  (a2),d0
        beq     putc

        moveq.l #'0',d1
ploop2: addq.w  #1,d1
        movea.l d0,a2
        move.l  (a2),d0
        bne     ploop2

        cmpi.w  #'9'+1,d1
        bcs     putc
        moveq.l #'+',d1

putc:   move.b  d1,(a1)+
        dbra    d7,ploop1

        move.b  #CR,(a1)+
        move.b  #LF,(a1)+
        sf.b    (a1)

        DOS     _PRINT
        addq.l  #4,sp
        rts
*
        .bss
        .even
*
hshtbl: .ds.l   HASHSIZ *ハッシュ表
*
        .text
        .even
*
*       メモリ不足
*
nomem:
        move.w  #STDERR,-(sp)
        pea.l   errms(pc)
        DOS     _FPUTS
*       addq.l  #6,sp

        move.w  #1,-(sp)
        DOS     _EXIT2

*
nfndms: .dc.b   '【新規】',CR,LF,0
fndms:  .dc.b   '【既存】',CR,LF,0
errms:  .dc.b   'ヒープを使い切りました'
crlfms: .dc.b   CR
lfms:   .dc.b   LF,0
*
        .bss
        .even
*
linbuf: .ds.b   2+256   *1行入力バッファ
*
heapst: .ds.b   65536   *ヒープ
heaped:
*
        .stack
        .even
*
        .ds.l   1024
inisp:

        .end    ent
```

卵はどちらの端から割るべきか

はずみで条件判断の「<」と「>」を逆にしてしまうといった類のバグは，なかなか気づかず，それでいて致命的な結果をもたらすという点でたちが悪い。恥ずかしながら『入門編』の単行本には，ビットの上位/下位と2進数での左/右の対応が入れ替わっているという間抜けな記述が今年の春先まで生き残っていた(それ以前の版を持っている人は45ページをみて笑ってやってほしい)。で，前回の原稿中，注1にもそのやってはいけない逆転バグがある。正しくは，メモリの若いアドレスに上位バイトを置くのがビッグエンディアン，その逆に若いアドレスに下位バイトを置くのがリトルエンディアンだ。以上，訂正するとともに，わざわざ指摘してくださった読者の方に感謝する。

ついでに補足しておくと，実はあの注(を上記のように訂正したもの)で述べているのは「日本での用法」であり，原義からはかなり逸脱している。そもそも，ビッグエンディアン/リトルエンディアンという言葉は，『ガリバー旅行記』に登場する「卵を大きいほうの端から割ることを主張する人たち(big-endians)」と「小さいほうの端から割ることを主張する人たち (little-endians)」に由来する。転じて，そのプロセッサが多バイトデータをメモリに格納する際に，値の大きな側(上位バイト)と小さな側(下位バイト)のどちらを先に(アドレスの若いメモリに)格納するかを表すのに使われるようになった。したがって，バイトの並び順(byte order)の“方式”ではなく，その方式を採用した“プロセッサ”，あるいはマシンなりメーカーなりを指すのに使うのが本来の用法だ。また，エンディアンという語を単独で用いるのも，おそらく日本の方言だろう。big-endianで1語，little-endianで1語であり，endianだけでは意味が通らない。いまひとつ自信はないのだが，元ネタを知らずにbig endianを「bigなendian」と読み違えたのが「裸の王様現象」を起こして広まったのだと思う。一時期の「PDS」や，最近の「FEP」の事情と似ていなくもない。

それにしても，言葉が足らなかったな，と気にしていたところに天の恵みの逆転バグ。期せずして訂正ついでに補足までできた。まずは，めでたしめでたし，だ(悪びれてない)。

追記：勘違いかもしれないが，「フリーウェア」も裸の王様現象を起こしている気がする。どなたか，この言葉の定義(と，できれば初出)をご存じであれば，無知な僕にご教授願いたい。

CREATIVE COMPUTER MUSIC

Creative Computer Music入門(15)

弦のアレンジ

今月からは，オーケストレーションのアレンジについて考えてみましょう。まず今回は，オーケストラのなかでも中心的な存在であるストリングスの利用法です。各々の楽器の音色の特徴や音域などをふまえたうえで，ストリングスらしいメロディラインを作ってみましょう。

Taki Yasushi 瀧　康史

§綺麗な曲を聴きたい

7：08AM。毎朝，静岡始発，東京行きの新幹線「こだま」に乗ります。その直後におそらく新大阪から来ている「ひかり」が静岡駅に止まり，東京駅に着くにはそれに乗ったほうが20分ほど早い。でも，あえてそれは無視（静岡←→東京間は「こだま」で約1時間半，「ひかり」で約1時間。料金は同じで，どちらに乗ってもかまわないので普通なら「ひかり」に乗るのが賢明)。

どおしてって？

後発の「ひかり」はなかなか座れなくてフルーツバスケットの状態だからとか？でもこの「こだま」に間に合う時刻に来るなら「ひかり」でも十分座れるし。この「こだま」は「こだま」なのに座りやすい新しい車両だからとか？　始発だから自分の好きなところに座れるとか？

そんなことは実はあんまり関係ないんですよね。なんていうのかな？　朝1時間半，お気に入りの席で，お気に入りの景色を眺めつつ，お気に入りの曲を聴いて，お気に入りの詩集を開き，お気に入りのパン屋のパンをたっぷり時間かけて食べて，物思いにふける。忙しそうでもちゃんと生活している人たちを見ながら行くのが，すごく好きなんですよね。

「ひかり」でもできなくはないんだけど。やはり，窓側の席に座れるのは稀な話だし。

そんなわけで，私はいつも「こだま」に乗って行きます。

さて。

このなかのお気に入りのCDってのが，クセモノ。こういう気分のときに聴くCDは，思いっきり美しくて，センチメンタルで，広がりのあるCDがやっぱりよい。冬だからかなァ？

でもってここで出てくるのは，やっぱりラヴェル。「逝ける王女のためのパヴァーヌ」とか，「古風なメヌエット」とかは有名ですよね？（古風なメヌエットはオーケストラ版がNHKの連続大河ドラマの時代劇でBGMとしてかかってたような気がする）彼の曲は作者本人がピアノ版，オーケストラ版を作ってますので，いろんな意味でアレンジの勉強になります。

ピアノがなぜ1台でオーケストラに匹敵する楽器といわれているか，最初にピアノ版を聴き，あとでオケ版を聴いてみて，あるフレーズがどんな楽器に置き代わっているのか。そういうことに気をつけて聴くと，実に勉強になります。

それに，ラヴェル独特の美しいメロディはいろんな人を魅了しえますから（すでに，いろんな人をとりこにしてしまったの)。「ラヴェル節」(笑）でも，学んでみてもいいかもしれませんね。

ちなみに私はピアノ曲では「夜のガスパール」のなかの1番目の水の精（ウンディーネ）のイメージがすっごく好きです。あれを聴いてると聴き込んじゃって，なんにもできなくなっちゃいます。

それから，オケ版はやっぱり「逝ける王女のためのパヴァーヌ」かなァ。「ラ・ヴァルス」とかも捨てがたいなァ……。

どちらにしても，美しい曲が聴きたい人は，ぜひぜひ買ってみてください（今回は強気の，人に薦めるモード)。

§ストリングス

ストリングスの使い方については，連載を始めたときから，いつかやってみたいと思っていました。通信で流れてくるデータを聞いていると，ストリングスを使いたくて使いたくてしょうがない人って絶対いっぱいいる，っていう自信があったから。

技術が足りなくて使い方を誤っている人。たぶん自分でもわかっていると思うのですが，いまいち納得できない人。1つひとつの音の音域がわからない人。そんな人はきっとたくさんいると思います。

ところが，これを説明するとなると，4声体（ヴォイシング）の知識が必要になってきます。でも，4声体は面倒くさいし，なによりも教えるのが大変です。理解するよりも教えるほうが大変かもしれません。それに奥が深い。

でも，そんなことよりも，うまく教えられるかどうか，自分にいままで自信がなかったこと(いまでも自信はないけれど)，それからもうひとつ。主にストリングスで構成されているオーケストラの曲は過去に2～3作ったことはあっても，まだまだ身についてないせいで，初歩的な間違いを犯すことがしばしばあること。いまでもありますが……。

それでも逃げてばかりではしょうがないし，自分がいまできる限りのことをお話しして，またあとで知識を得たら，そのときにまたお話ししようという開き直りから，今回はこれをやることにしました（で，先月4声体をやったのです)。

まあ，私はこれだけの知識で30～40分程度のものを2～3曲作ってしまったわけですが，実際に私が演奏できる楽器はピアノだけで，いまさらほかの楽器を使えるようにはなれないでしょうし，オケの曲を作れる人が必ずしも，すべての楽器を使えるわけでもないでしょう。そんなわけで，弦楽器を主に使っている人で，「私の楽器でこんなことはできん。コイツはおかしいぞ」ということがあったら，ぜひ私宛にお手紙ください。お礼はできませんけど（図々しい……)。あ，名前公開可か，不可かは書いてくださいね。

雑談はさておき，まずはよく使われるストリングスについてお話ししていきましょう。

まず，ストリングス……ここでは，ヴァイオリン，ヴィオラ，チェロ，コントラバスの4つをイメージしてください（ストリングスというと，一般的にはさらにハープが入って5つになります)。

どの楽器も，大きさを除いて形は酷似していますよね。弦が張ってあり，その弦を弓で擦ったり(arcoアルコ)，指で弾いたり(Pizzicatoピチカート）して音を出すのです。

どれも弦は４本あります。ギターなどを知っていればわかるでしょうが，この弦を押さえて音階を変えます。押さえることによって響く部分の弦の長さが短くなり，音が高くなるわけです。

ここで，まず覚えなくてはならないことは，この４つある弦（調弦という）はなにも押さえなければどの高さの音が鳴るかということ。当然，一番太く低い音が出る弦をどこも押さえていない状態で出る音が，その楽器の一番低い音です。

ストリングスを使うときには，この調弦の音がどこなのかも意識して使わなくてはなりません。DTMでシミュレートするならこの調弦を覚えておけば，よりよい使い方がわかるはずです。

主な４つのストリングスの特徴と，調弦，音域などを以下に記しておきます。

○Violin（ヴァイオリン）

ストリングスを使うといえば，まずこの楽器だといってよいほどメジャーな楽器です。

ヴァイオリンは，ストリングスのなかでは最も小さい楽器です。弾くときは，肩にのせてあごで押さえて固定しますが，この弾き方は，次に説明するヴィオラも同じです。弦の長さが短いため，音域は最も高く，繊細な音が出ます。

オーケストラのメロディを奏でるときもあれば，ハーモニーを奏でるときもあり，音域も広いし，音自体に表情があり，柔らかい美しい音を出したり，寂しそうな音を出したり（ドリフなんかでヴァイオリンが人の声を真似るやつとかもあったでしょ），まあ，そんなわけで，用途が広く，かなりよく使われる楽器なのです。

オーケストラのなかでは，音が小さいわりにソロなんかでもよく使われ，いろいろなソロシーンがあります。R. コルサコフの「シェエラザード」なんかは，最初からむせび泣く哀愁のメロディが奏でられてますよね(これはカッコいいぞ)。

調弦はO3G，O4D，O4A，O5E。音域はO3G～O7Eです。

○Viola（ヴィオラ）

ヴィオラは有名ではないのか（十分有名だと思うのですが)，なぜかCM64-5版カードに入ってません（不満の声……)。

ま，そんなことはどうでもいいんですが，楽器の形状は，ヴァイオリンをそのままちょっと大きくしたような形です。しかし，音色はただヴァイオリンを低くしたのではなく，ヴァイオリンと比べるとアタックの弱い，マイルドで淡い響きがあります。使い方は，ヴァイオリンが主に主旋律（メロディ）を奏でるのであれば，ヴィオラはオブリガードに使ったり，主旋律であるヴァイオリンから５度下でハモるとか，そんな使い方が合うでしょう。O3がすべて出るので，金管楽器などとハモるのもよいかもしれません。

さて，忘れてはならないのは，ハ音記号の楽譜（図１）です。Cの反対みたいなのが２つついている記号なのですが，実はコレ，ヴィオラのための楽譜なんですね。

この読み方ですが，５線上の第３線，真ん中の音がO4Cです。これは，ヴィオラの旋律がたいてい，O3～O4をいったりきたりしているため，ヘ音記号で書くと上に加線が増え，ト音記号で書くと下に加線が増えるためです。要は見やすくしたわけです

しかし，このCreative Computer Music入門では，この記号は使いません。それでなくても音楽記号は多いのでややこしくなりがちなため，ヴィオラはヘ音記号で┌8va――（オクターヴァ記号）をつけることにします。こうすれば，ハ音記号の楽譜より１つだけ下になりますので加線も少なくなります。

ハ音記号は「真ん中がO4C」と覚えておけば，DTMだけをやってる人には十分です。これだけ知っていれば，そういう楽譜があってもMMLに直せますものね。

調弦はO3C，O3G，O4D，O4A。音域はO3C～O6Dです。

○V.Cello（チェロ）

先頭のVはViolaの略です。短縮してチェロといいます。ヴァイオリン，ヴィオラなどが肩で支えて弾くのに対し，このチェロは足の間にはさみ，床に立てて演奏します。

大きさも結構あるので単体でもよく響き，音に厚みがあります。音のヴォリューム自体も弦楽器のなかでは最大です。

また，音域も十分広く，かつベースノートからトップノートまで押さえることができるため，チェロひとつだけでかなりの曲を弾くことができます。私の好きな無伴奏チェロなどもそうですが，チェロの音は非

図1　ハ音記号

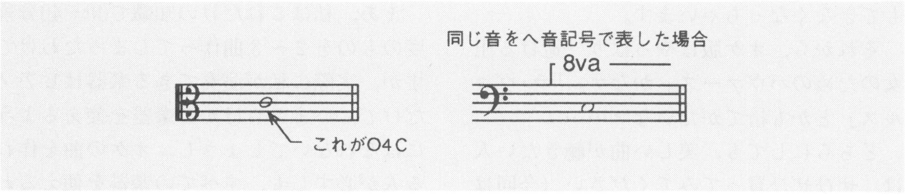
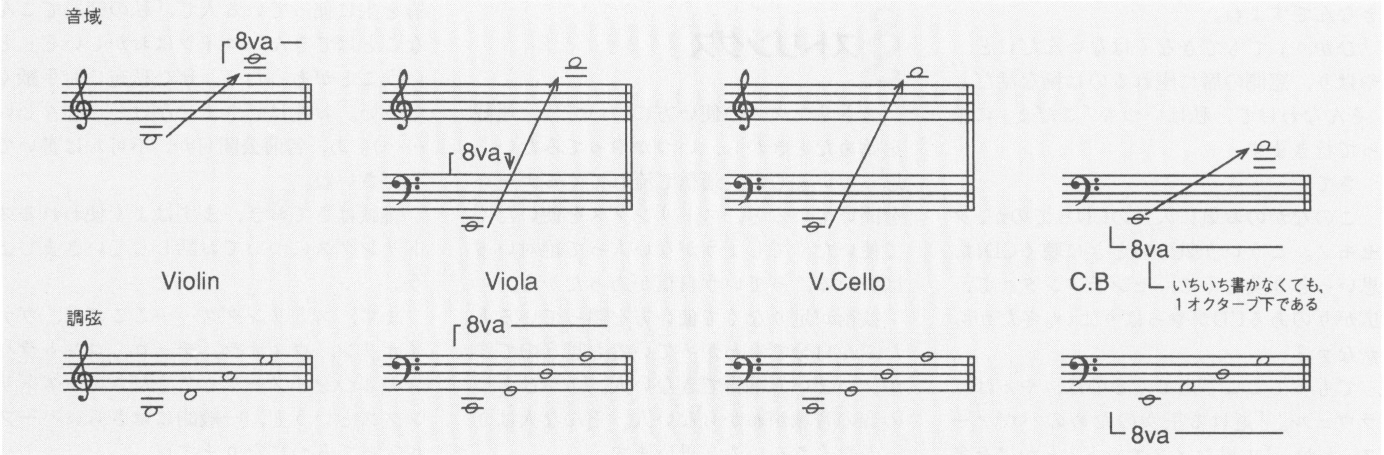



図2　各楽器の音域と調弦



常にあたたかく（ヴァイオリンのように攻撃的な音は聴いたことはありません），滑らかで非常に美しいのです。これはおそらく，音域が人間にとって最も聴きやすい音域であるためでしょう。

調弦はO2C，O2G，O3D，O4A。音域はO2C～O5Bです。

○Contra Bass（コントラバス）

コントラバスは，チェロをさらに大きくした形をしています（輸送は大変そう）。

そのため，非常に低い音が鳴りますが，あまりに低いため人間の耳ではほとんど聞こえません（おそらくボリューム自体は非常に大きいのだと思います）。

正確な周波数はどの程度かわからないのですが，これを鳴らすとなると，その辺に転がっている安いスピーカーじゃ鳴りっこありません。

コンバス（コントラバスの省略）は生演奏じゃないと，まともには聴こえない，いや，体で感じることができないってのが正確な言い方でしょうか。

実際，曲を作るときにコンバスの音を正確な音域で鳴らすと，私が愛用しているモニタについているスピーカー（CU21HDだから，たぶんCZ-614Dなどについているものと同じでしょう），あの耳みたいなやつだと，ほとんど聴こえません。そこらによくあるCDラジカセなどについているものでも，よく聴き取れるくらい満足できる音にはなりませんでした。ちなみに，インナーホンでも同様。1万円弱する両耳に押し当てるヘッドフォンならば，採譜できるくらいには鳴るけど，ちょっとね。

結局，満足できる（体で感じられる）音で鳴るのは，私の家ではダイアトーンの100Wスピーカーぐらいでした。

コンバスは，四重奏（ピアノ，ヴァイオリン，ヴィオラ，チェロ）でも省かれますし，そもそもチェロが十分低い音が出るため，ベースノートをしっかり奏でられるので，DTMではあまり意識しなくてもよいでしょう（どうしても使いたければ別ですが）。

調弦はO1E，O1A，O2D，O2G。音域はO1E～O3Aです。

これらの楽器の音域と調弦を図2に記しておきます。

> 注　「バナナパフェ味のそよ風」で使われているコンバスは，正確な音域ではありません。あれは，コンバスのピチカートの音がなかったこと，普通のCARD5のピチカートの音域を下げたら変になってしまったことから，1オクターブ上げています。実際にはチェロ2本で，1つがピチカートで奏でると考えるほうが正確でしょう。
>
> ここでいっているのは，オーケストラをCDで聴いて実験したものです。

§和声と照らし合わせる

まずは図3を見てください。これは先月にも掲載した4声体の各和声部の音域です。これと図2を照らし合わせてみて，ストリングスという楽器で4声体を作ると，C(maj)の開離配置ならば図4に示されるとおり，5つになります。

先月の範囲なのですが，仮に，音がたったの4つしかなくて，4声体の成分を作るのがやっとだったとします。そのときの開離配置の特徴は，密集配置に比べて，和声が広がりをもって配置されているため，ハーモニーに広がりがあります。

連載中で何度もお話ししたとおり，人間の声というものには実にさまざまな周波数帯の音が含まれていて，非常に厚みがあります。そのため，人間の声ならば開離配置でもなんでもかまわないのですが，楽器の音の場合は非常に問題があります。

ストリングスでやる場合も同様です。

生のストリングスであれば，音には十分すぎるほど厚みはありますので，開離配置にしてもよい（というか，音に厚みがあるため開離配置が望ましい）のですが，シンセサイザの音の場合は，音色を考えたうえで，場合によっては密集配置にしなくてはならないかもしれません。

ちなみに，私はCM-64のストリングスは，開離配置にすべき音だと思っています。

次回で説明することですが，ブラスのオーケストレーションでは（たとえ生であっても）密集配置にしなくてはなりません。金管楽器（ブラス）の音は正弦波に近いため，コードにすきまができると，ハーモニーにスキができやすいためです。

密集配置は開離配置に比べてコードの構成音にスキができないので，オクターブユニゾンなどでコードに幅をもたせます。

余談ですが，4声体でうまく配置された曲に四重奏というものがあります。実は，この連載の最初のほうに紹介したCD「Pre・Primer」1，2は全曲，この四重奏という形をとっています。

この楽器の構成は，ピアノ，ヴァイオリン，ヴィオラ，チェロで，ここであげたストリングスのうち，コンバスが入っていません。これは，チェロが肉声のバスを包含するほどの低音域を網羅しているためで，このことは，コンバスはなくても十分美しい4声体を保つことができることを物語っています。

§ストリングスらしいメロディライン

前章ですでに，和声の面からみたストリングスの使い方をお話ししましたので，ここではストリングスでの，美しくかつストリングスらしいメロディラインの作り方をお話ししましょう。

図5を見てください。このメロディは編集室の某人のX68000 XVIの前で，某4文字のゲームでSで始まってNで終わり，最後に3がついているワイヤーフレームゲームのデモらしきものを眺めていたときに思い浮かんだメロディです。まあ，最初からストリングス＋木管楽器のイメージで作ったメロディなんですけど（いま，それを形にしようと22段譜に書いてる途中）。

で，図6は，このメロディにストリングスらしい装飾音をつけてみたものです。とりあえず，楽器はヴァイオリンと限定しておきましょう。まあ，ここでの楽譜は教材ということで過剰なほど装飾音がついています。ですから，実際に曲ができ上がってしまったら，ここまで装飾はつかないでしょう。

図3　各声部の音域

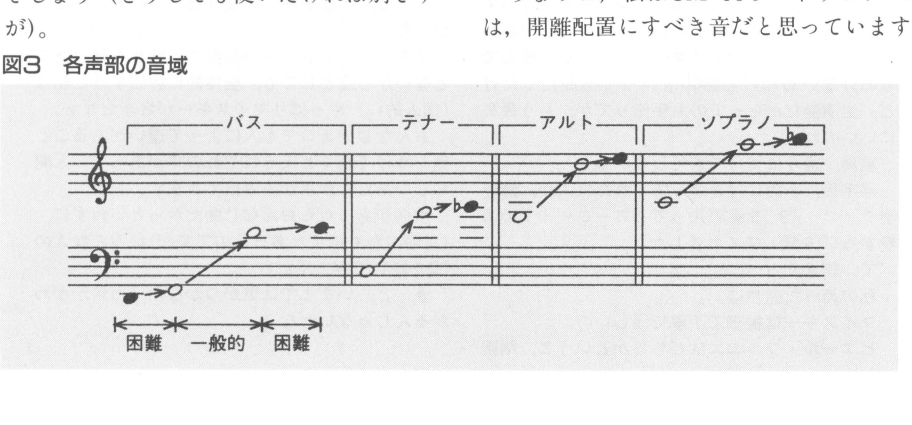
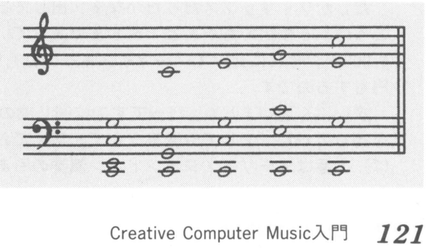



図4　C(maj)の開離配置



では最初に。１小節はポルタメント。「polt.」と書いてあります。ポルタメントは一時期DTMの，しかもMML地方で流行ってましたから，どのような記号かはすぐにわかることでしょう。

ここでは短２度下のB♭から滑らかに上がります。ストリングスのポルタメントはたいてい，短２度から上がったり下がったりします。注意しなくてはいけないことは，上がる前の音にアクセントをつけることです。これは絶対に忘れてはいけません。変に，「らしく」作ろうとして逆に失敗しているデータのほとんどの場合は，ここにアクセントをつけていないのです。アクセントをつけないと，ただにゅ～っと音が上がって気持ち悪いだけになってしまいます。

え？　Z-MUSICではどうやるかって？　うぅ……知らん。善ちゃんのことだから，このくらいは考えてあるに違いないので，質問箱にでも出してみましょうか。

次。２小節目の重音。当然，弦の数は４つなので最高４重音になりますが（ただし，ギターと違い，弦の張り方が水平ではなく，丸みを帯びた弧の形をしているので完全な重音になるのはやはり２重音ぐらいでしょうか。それ以上になるとディレイがかかってしまいます），ストリングスでは場合が限られます。

注目すべきことは片方が開放弦（調弦のままフレットを押さえずに演奏）であること。これらがもし，開放弦でないとすれば，厳密にやるなら隣り合った弦の音の差をふまえて，押せるようにしなくてはいけません。

そうそう，余談としていっておきますが，もしこの２音にポルタメントをかけるとしたら，この場合開放弦にはかかりません（理由はわかるでしょう）。また，この開放弦にはビブラートなどはかけられないので，念のため。ちなみにトレモロは，通常とは違う弓を使ったトレモロとしてできるそうです。

メロディラインの下や上などで重音で演奏するときは，片方は動かず開放弦にしてしまうか，そうでないにしても，３～４度ぐらい音の間隔に幅がないと演奏は困難になります。

もっとも，ストリングスの演奏をそこまで正確にシミュレートするかどうかにもよりますが。

その直後の音符についているのがヴィブラート記号（vib.）です。これは単純にヴィブラートをかけるだけですので，たいした問題ではないでしょう。ただ……音源ドライバなどについているサイクルが一定のLFOだとあまり美しくありません。単にうねうねしてしまうだけで，いまいちであることだけはいっておきましょう。

３小節目の頭についている記号は，装飾音の指定です。次の音を鳴らす前に心持ちこの音を鳴らします。よく，これを16分音符２つにしてしまう人がいますが，それは間違い。これは装飾音であって16分音符ではないのですから。タイミングとしては，前の音を早めに切り，１：５から１：３ぐらいの割合で鳴らすと，それっぽいでしょ

図5　ゲームのイメージで作ったメロディ

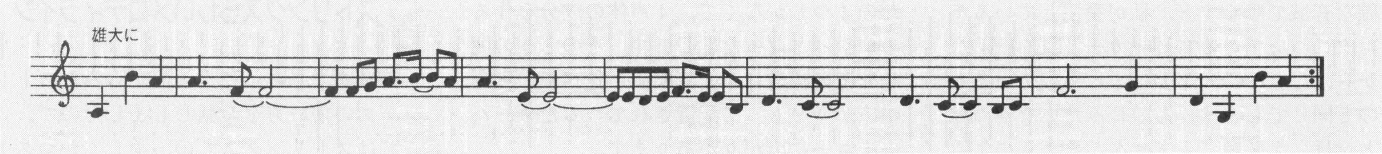
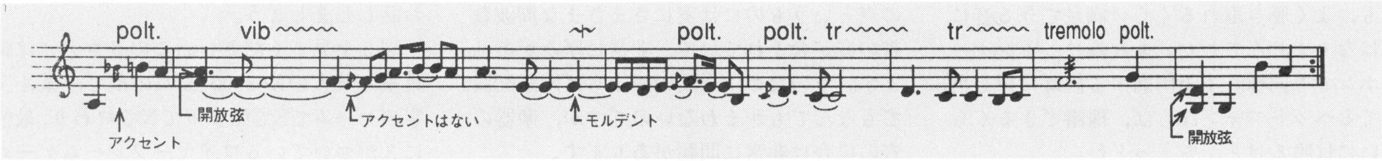



図6　ストリングスらしい装飾音をつけてみる

マイスキーのコンサートに行ってみた

先日，ミッシャ・マイスキー（チェリスト）のコンサートに出かけてきました。私がチェロが好きなのはみなさんも知ってのとおり。でも，なによりバッハの無伴奏チェロ３番を演奏するから，行きたい！って思ったんですよね。

生で……しかもソロで聴くチェロの音は，やっぱりCDなんかよりもずっとよかった。体で感じることができたから。当然席はS席で前のほう（結構すみのほうだったけど）。会場に着くと例によって例の如く，マイスキー演奏の無伴奏チェロ組曲（バッハ）１番～６番の３枚組の全集を売っていたんです。

実は予習も含めて無伴奏チェロ１～６の全集を，すでにピエール・フルニエの演奏するCDを持ってたんですよね。

たしかフィリップスだったかな？　出してるところは。２枚組でちゃんと１～６まで入っていて，4,000円。それに比べて……マイスキーのは7,500円もするのです。

ずいぶん迷いました。だってすでに同じ曲のCDを持ってるし，ましてや全集なんて。考えてみれば，１番はモーリス・ジャンドロン演奏のもあるし。いくらチェロが好きだといってもチェリストじゃないんだから，いくらなんでもなァって。最初はやめにしたんです。

でもね。演奏を聴いて（無伴奏以外の曲は何が出てくるかもわかりませんでした），繊細でかつ，あたたかい彼のチェロを聴いて。ああ，やっぱり人によって同じ曲でも解釈が違うんだな。といまさらながら感心。

その日の公演で演った曲はアンコールは除いて（アンコール５回もやってくれたの。ありがとね……といっても，マイスキーさんこんなの読んでるわけないわね）全部CDを持ってる曲だったけど，生演奏だからってのも手伝ってか，もう最高にいいのね。

結局，帰りに買ってきてしまいました。

帰宅後，一緒に付き添ってくれた友人が，無伴奏チェロ１，３，５番の入ってるヨーヨー・マの演奏するCDを貸してくれました。

で，聴きくらべ。

私の持った感想は。

マイスキーは繊細で丁寧で優しいの。

ピエール・フルニエはどちらかというと，剛腕のお父さんの力づよい優しさって感じ。

ヨーヨー・マは，機械的な硬さがある。私はチェロっていうと，暖炉のようなあたたかさを思い出すけど，彼のチェロはなんか無機質で冷たい感じがしたな。私はあんまり好きなタイプじゃないみたい。

モーリス・ジャンドロンのは，マイルドな優しさかな？　これはチェロ１番しか聴いたことないんだけど，１番を聴いた限りだとこの人が一番よかったな。

まあ，ジャンドロンの演奏のはほかを聴いたことないから別としても，全体的に評価すると私は（個人的に）やっぱりマイスキーが好きだなあ。

おんなじチェロでも人によって違いがあること。気をつけて聴くと人それぞれの味があって，人間らしいあたたかさが染み出てきます。

機会があったらおんなじ曲だからといわずに，（実際，私も抵抗があったのですが）いろんな人のCDを聴いてみてください。

きっと，いままでは気がつかなかった何かがわかるんじゃないかな？

う。

音符の下（もしくは上）についているギザギザの記号（6小節目）はトリルです。このギザギザマークに縦棒が1本入ったのが，モルデント。

トリラーは短2度（指定がある場合もある）上と交互に8～16分音符の速さで弾き，モルデントは逆に短2度下と交互に弾きます。トリル，モルデントは図7を参照のこと。実際にはどのような感じで演奏するかを書いてみました。ただし，この音符の長さや数は単なる目安で，実際にどう演奏するかは個人の好みによるので，データに直すときにはじかに直すとおかしくなります。

最後の小節は，これもまた重音になっています。ここではたまたま2つの音が開放弦でしたが，実際の規則では，1つだけ開放弦にするという規定のみです。これはアクセントをつけて鳴らしてください。

図6では出ていませんが，ストリングスの特殊な弾き方で有名なものに，さっきいったピチカートというものがあります。まあ，弦をただやさしく弾くだけですが（ちなみにチョッパーベースのように，弦をビン！と弾いてしまうものもあります），ここで注意しなくてはならないことは，ピチカート←→アルコの切り替えですね。

アルコが弓で演奏するのに対して，ピチカートは弓を手に持って，指で演奏するわけですから，切り替えには多少の時間がかかります。だいたい4分休符～2分休符ぐらいは弓の持ち替えの時間を入れたほうがそれらしくなるでしょう。

また，ピチカートではそれほど速く演奏できない（16分音符ぐらいがやっとだな。曲全体のテンポにもよるだろうけど）ので，曲を作るときは気をつけなくてはならないでしょう。

§ストリングスを使って盛り上げる

ストリングスというのは実際，華やかな楽器です。オーケストラなどでも，ここいちばん，華やかにカッコよく盛り上げたいときなんかによく使われるテクニック（というほどでもないんですが），駆け上がりというのがあります。

で，ピンとこない人のために駆け上がりというのはどういうものか説明しますと，図8に示したようなものです。言葉で説明してしまえば，単に音を順々につなげていくだけのものです。

たいていは，盛り上がらなくてはならないとき（具体的な場所でいえば，テーマの直前やサビの前など）に使われたりします。でも，実際使われているときはたいてい速すぎて，カッコいいのでマネてはみたいけれども，いったいどのようなことをやってるのかはサッパリわからないなんて人も結構いるでしょう。

さて，いってしまえばグリッサンドのようなもんですから，ここで重要になってくることは次の3つに絞られます。

1)　どの音から始まるのか？
　　(始まりのコードはなにか？)
2)　どの音で終わるのか？
　　(あとに続くメロディはどの音から始まるか？ また，どのようなコードか？)
3)　どのくらいの長さがあればよいのか？
　　(音符はいくつぐらいがよいか？)

駆け上がりは，うわ～っと上がってしまいますが，勢いで上がるのではなく，ちゃんと和声的な法則にのっとって駆け上がります。

まずは図9を見てください。1小節目のコードはG7，2小節目がCです。2小節目からこの曲のテーマで，その前のG7(G7というところからも）ではCを思いっきり盛り上げたいところです。

最初は単純に，まず終わる音はCです。始まる音は決めていません。ただし前のコードはG7です。長さは短めに1拍ということにしましょう。

駆け上がりは8分程度の長さの音符がたいていちょうどいいので，1拍では4つになります。しかし4つというと，到着点のCから数えて，同一スケール上ではB，A，G，Fと，Fまで下がってしまいます。このままだと駆け上がりの始まりはFになってしまいますよね。

けれどもコードを見ると，ここはV7(G7)駆け上がりの場合，最初の音には結構インパクトがあるので，必ずといっていいほど，コードの構成音である必要性があります。あります。

そこでうまく考え，最初の音をうまくGにするように訂正する必要があります。この場合，簡単に考えられるのは3つほどで

図7　トリルとモルデントの違い

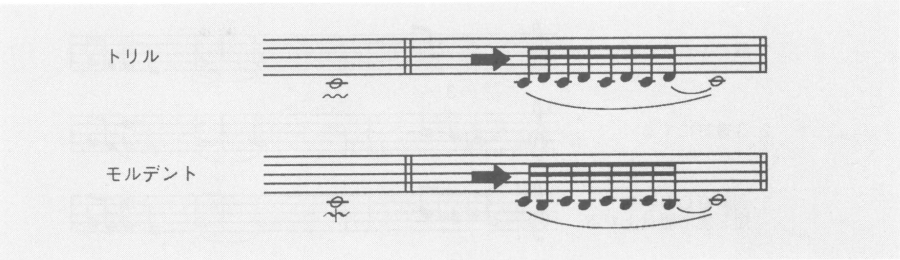
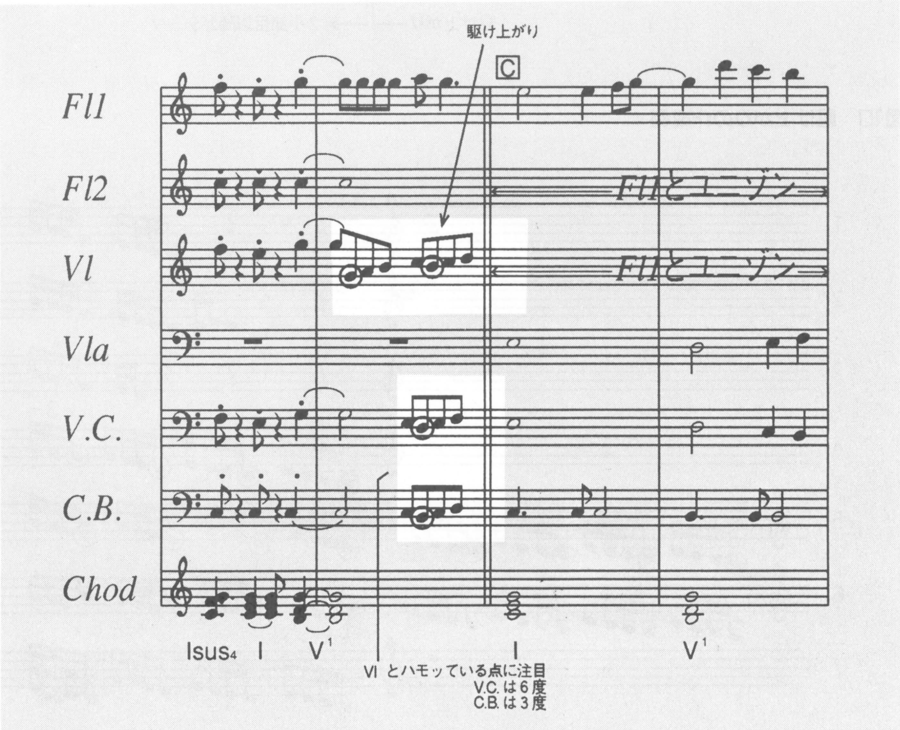


図8　駆け上がりの例

す（図9-1，2，3）。1つは，最初の音は休符にしてしまう方法。2つ目は3連符にしてごまかす方法。そして最後に，4つ上げる代わりに，到着音Cの一歩手前のあたりで短2度で駆け上がるという方法。

どちらかというと，2が一番メジャーです。1もしばしば使われますが，3はあまり使われていません（といいつつ私はこういうのが好きだったりします）。

長くなればなるほど，複雑になりますが，たいしたコトではありません。同じメロディテーマに続ける駆け上がりをいくつか作ってみました（図10）。

まず，1はオーソドックスに16分で上がる型。注目すべき点は最初の音がB（G7の構成音）。

2は装飾が2拍目についただけ。

3は，Bではなく，その上の構成音Dから駆け上がるパターン。

4は，○で囲まれている点に注目。この音がコードの構成音を踏みつつ上がります。

5は4と似ていて，○で囲まれている音符が徐々に上がるパターン。

6は，5に味つけをしていて，○のついた点から先は16分音符の分だけずれている例。この駆け上がりは長いので，単純化を避けるため，また，最後の1ブロックの先頭が，5ではFから上がるが，6ではGから上がるという点などを工夫。

7は4を改良して，最後の1ブロックだけ重音している例。

基本的には駆け上がりは混乱を防ぐために重音ユニゾン程度に避けたほうがいいのですが，ここでは厚みをもたせるために，6度の重音をしています。

5度，4度，6度などは，比較的綺麗につなげるので有用でしょう（ただし，最後の1ブロック程度にしたほうがよい）。

短いですが，実用の例として，図8は「バナナ」のサビの直前の盛り上げ部分です。囲んであるところが，駆け上がり部分ですが，ヴァイオリンに対して，チェロ，コンバスが6度，3度（これはずいぶん低いですから）ハモっていることに気がつきましたか？

この例でも，○がついている音はコードの構成音です。

§まとめ

私は四重奏が好きで，しょっちゅう聴いています。ピアノとストリングスの絡みあいが，すごくよい味を出していて好きなのです。

だから，最初に弦楽器を勉強したのはやっぱり四重奏を作りたいからでした。

実際に楽器を弾く人でも，すべての楽器をマルチにこなす人はいません。しかし，自分が弾けない楽器だからといって遠慮してしまうと，多彩な曲は書けません。

せっかくDTMでそれらしい音を出せるのですから，いろいろ試行錯誤して，面白い曲を作ったりアレンジしたりしてほしいと思います。

さてと。来月はブラス（金管楽器）をやるつもりですが，実はブラス楽器を使うのは苦手なので，あと回しにするかもしれません。情報を仕入れなくちゃいけないから。

親切な読者の方がネタを提供してくれました。「オープニングやエンディングの作り方」とか，「メドレーの作り方」などいろいろ。どうもありがとう。

そんなわけで，これらについてもそのうちやっていきたいと思います。

そうそう，みなさんからメロディのつぎはぎを募集して，それを組み合わせて曲にしてみよう，という企画も面白いかもしれませんね。

それではまた来月。

図9　駆け上がりのパターン

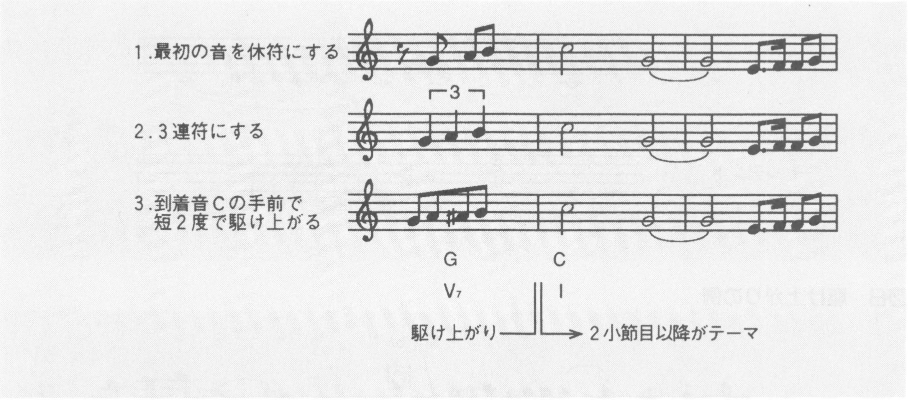
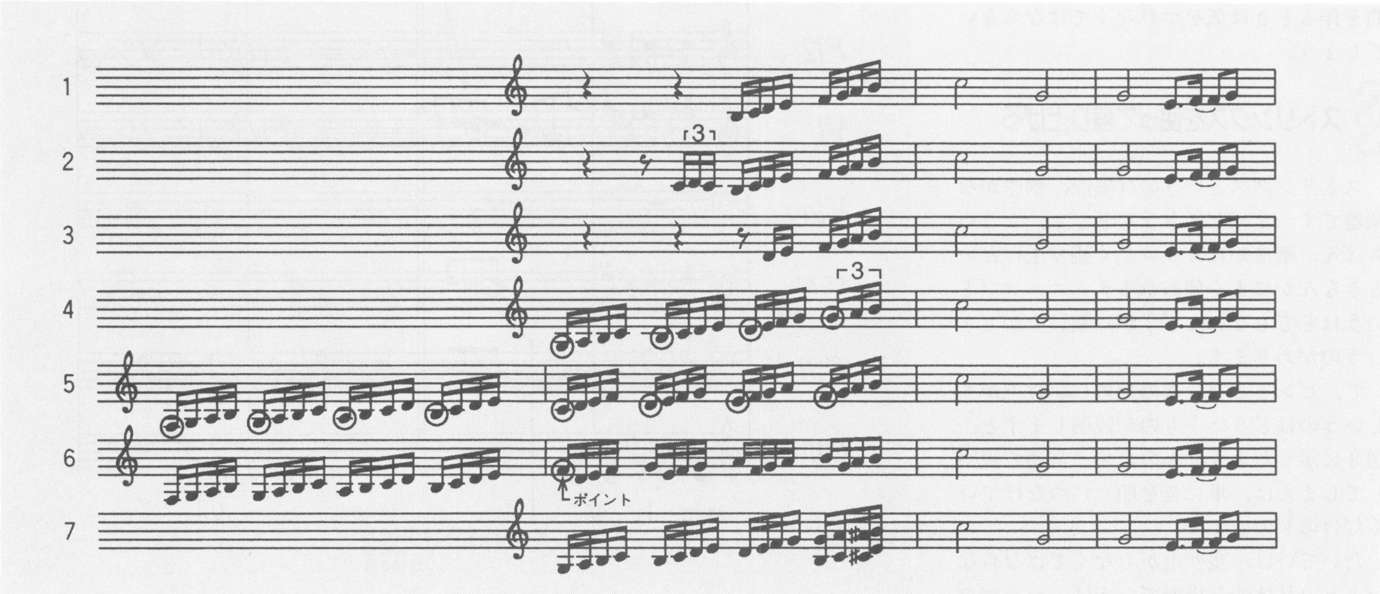



図10　駆け上がりの作成例

Oh!X

LIVE in '92

X68000・Z-MUSIC用
(要MT-32同等品)

LAST CHRISTMAS

Endo Ryuichi 遠藤 隆一

X68000 ©システムサコム

闇の血族より 次回予告のテーマ

Ono Mikio 小野 美樹夫

X1・musicBASIC用 ©システムサコム

ユーフォリーより オープニングテーマ

Nisio Masato 西尾 将人

この季節，レコード屋さんにはさまざまなクリスマスソングが並びます。LIVE in '92の締めくくりの1曲目も，12月らしく「LAST CHRISTMAS」です。そして，ミュージッククリエイター斎藤学さんの追悼として2曲を紹介しましょう。

あの頃，ラストクリスマス

今月のX68000にお届けする1曲目はZ-MUSICシステム用の洋楽です。ジョージ・マイケル，アンドリュー・リッジリーといえば泣く子もだまる「WHAM！」(ワム！)の2人組ですよね。WHAM！は今から10年ほど前に「WHAM RAP！」でデビューし，「YOUNG GUNS」で瞬く間にビルボードのヒットチャートを上り詰め，その後も「BAD BOYS」「FREEDOM」「CARE LESS WHISPER」などのヒットを連続して出しました。そして忘れてならないのは今回紹介する「LAST CHRISTMAS」です。今でもクリスマスシーズンになるとレコード屋にはCDが並び，ラジオや有線などでも流れています。“名曲は滅びず”ってやつですかね。現在，WHAM！は解散していますが，ジョージ，アンドリューとも音楽関係の仕事をしているので，どこかで2人がやった音楽を聴いたことがあるかもしれませんね。

すっかり導入が長くなってしまいましたが，MT-32同等品が必要ですので，注意してくださいね。

さて，リストを見ると，果てしない長さがあるようですね。過去の例からもわかるように，長いリストでも掲載する価値があると判断されたわけですので，曲のデキは保証されています。

演奏前にLA音源をエディットする必要があります。音色リストを入力したあと，LHA.Xで展開してください。さらに展開したファイルを，

COPY 音色ファイル MIDI

としてから演奏を開始してください。

この作品の投稿は10月の初旬にあったわけですが，実にうまいタイミングを見計らっています。遠藤君はクリスマス関係の曲をメインにした投稿をしてくれたのですが，季節物の投稿は3カ月ぐらい早めに出すのがベストだといえます。

この作品は掲載の関係上，リストを若干修正しています。もちろん，スピーカーから奏でられる作品自体はオリジナルのままです。

闇の血族

さて，X68000用の2曲目をお届けする前に悲しいお知らせがあります。「対談・ゲームミュージックコンポーザー」のコーナーの第1回目(91年7月号)に登場していただいた斎藤学さんがお亡くなりになりました。

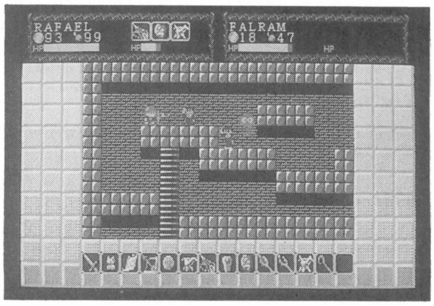
そこで，このコーナーでは追悼として斎藤さんの作品を掲載したいと思います。

その1曲目となるのが，システムサコムから発売されていた「闇の血族」の「次回予告のテーマ」です。「闇の血族」はノベルウェアと呼ばれるアドベンチャーゲームであり，前編・後編に分けられていました。その前編のエンディング(?)で流れていた曲です。

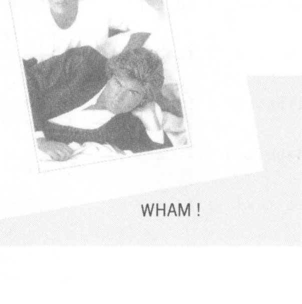
この作品自体の投稿は2年ほど前になり，おそらく投稿してくださった小野君は確実にボツだと思っていたことでしょう。実は，掲載用のストックに入ったままになっていたのです。

演奏はX68000本体のみで可能です。サンプリングは使っていないのですが，ZMUS

闇の血族



ユーフォリー



WHAM！

IC.FNCを使っているので注意してください。Z-MUSICシステム用のMUSICZ.FNCではありませんよ。

久しぶりだよ，ユーフォリー

X1用でも斎藤さんの曲をお届けしましょう。斎藤さんといえば「ユーフォリー」，「ユーフォリー」といえば斎藤さんというくらいの代表作です。曲自体は以前にも掲載されたことがありますが，今回はMIDI対応ということになっています。

作品は「ユーフォリー」の「オープニングテーマ」で，U-220用です。

X68000ユーザーが多くなった現在では「ユーフォリー」を知らない人も増えてきたかもしれません。「ユーフォリー」はシステムサコムから発売されていたRPGで，独特の雰囲気と高い音楽性で話題になったゲームです。音楽などは今でも名作のひとつにあげられるほどのものだったのです。

プログラムでの注意点としては，Vコマンドしか使っていないのでベロシティしか変化せず，ボリュームは設定していないので，初期化しないとバランスがくずれるかもしれないということです。なお，音色は工場出荷時と同じに設定してください。

最後になりましたが，斎藤学さんのご冥福を心よりお祈り申し上げます。　(S.K.)

リスト1　LAST CHRISTMAS

日本音楽著作権協会（出）許諾第9271868-201号

```
////////////////////////////////////////////////////////////
/                                                          /
/   L A S T   C H R I S T M A S       W H A M !            /
/                              プログラムＥＤ　えんりゅ　   /
/   L A   S O U N D   M O D U L E   (除：Ｄシリーズ)        /
////////////////////////////////////////////////////////////
(I)
(B0)
(M1,5000)(AMIDI1,1) /エクスクルーシブ
(M2,5000)(AMIDI2,2) /ボーカル
(M3,5000)(AMIDI3,3) /ヴォーカル
(M4,5000)(AMIDI4,4) /S Y N T H
(M5,5000)(AMIDI5,5) /エレピ
(M7,5000)(AMIDI7,7) /E. B A S S
(M9,5000)(AMIDI9,9) /イントロエレピ
(M20,5000)(AMIDI6,20)
(M21,5000)(AMIDI8,21)
(M10,5000)(AMIDI10,10)
(M11,5000)(AMIDI10,11)
(M12,5000)(AMIDI10,12)
(M14,13000)(AMIDI4,14)
(M16,5000)(AFM1,16)
(M17,5000)(AFM2,17)
/S E T U P   L A音源//////////////////////////////////////
.roland_exclusive 16,22 ={
                $10,0,1
              0,4,5,
              2,2,16,4,0,2,0,0,6
              1,2, 3,4,5,6,7,8,9}
/F M   V O I C E//////////////////////////////////////////
(v1,0,52, 15, 0, 0, 0, 0, 0, 0,0,3,0,
        31, 13, 0, 0, 4,10, 0,10,7,0,0,
        16, 14, 6, 5, 8, 7, 0, 2,7,0,0,
        31,  0, 0, 0, 0, 0, 0, 5,3,0,0,
        18, 13, 6, 5, 8, 7, 0, 2,3,0,0)
/I N T O R O//////////////////////////////////////////////
(T1) T86
(t9) R1@15@v085r8 o4q8L16.z80,90,82,75          '>a2.<c+e',2 |: 'c+8.e' :|  '>b2.<d'
(t9) @d1'>ab<d',2  >b<|:d>b<:|@d0  'c+2.e' @d1'>a8.<c+e' 'd8.f+''c+8.e'  @d0'>b2.<d'
(t9) >df+ab<de  f+2g1  '>b-8.<dg',2  'f+8.a'b-4 >a2.<  T80 'e1a'
(t20)  r4@14r2@v30r4o4q8L1  R8@d1 'e1a'r1 |:12r16.:| r1r8.r2. @d0
(t20)  q6'>b-2<df+'q8        @d1'>b-1<deg'r8.r8.r4@d0 '>a1<ea'&'>a2.<ea'
(t21)  r4@14r2@v25r4  q8o4  r8@d1'c+2.>a<'r8.r8.   @d0 @d1'd2.>b<'|:6r16.:|@d0
(t21)                              @d1'c+2.>a<'r8.r8.r8.@d0 @d1'd2.>b<'|: r16.:|@d0
(t21)  r2..r1g8.a8.b-4  d2.c+1
/ベース////////////////////////////////////////////////////
(t7) r2@v127@u80r1 o2q8L8@p64r1 @65r1 |:7 r1 :|
(t7) t120 |:12 d :|  c+dc+d  >|:14 b :|  <d>b<
(t7) |:12 e :|  f+gf+g  >|:a4aa:|  a4b4<c+4e4
(t7) |:12 d :|  c+dc+d>  |:8 b :|  b4f+4b4<c+d  |:12 e :|f+gf+g
(t7) a4.>a2a  a4b4<c+4e4
(t7) |:12 d :|  c+dc+d>  |:8 b :|  bbbbbb <c+d  |:12 e :|f+gf+g
(t7) >|:8 a :|  a4b4<c+4e4
(t7) |:12 d :|  c+dc+d>  |:8 b :| b4<f+4>bb<c+d |:12 e :|f+gf+g
(t7) a4.>a2a<  agf+gf+edc+
(t7) |:12 d :|  c+dc+d>  |:8 b :| b4<f+4>bb<c+d |:12 e :|f+gf+g
(t7) >|:8 a :|  a4b4<c+4e4
(t7) |:12 d :|  c+dc+d>  |:8 b :|  bbbbbb <c+d  |:12 e :|ee f+g
(t7) aa4>a2a<  a4g4f+4e4
(t7) |:12 d :|  c+dc+d>  |:8 b :|  b4f+4b4<c+d  |:12 e :|f+gf+g
(t7) a4.>a2a  a4b4<c+4e4
(t7) |:12 d :|  c+dc+d>  |:8 b :|  bbbbbb <c+d  |:12 e :|f+gf+g
(t7) >|:8 a :|  a4b4<c+4e4
(t7) |:12 d :|  c+dc+d>  |:8 b :| b4<f+4>bb<c+d |:12 e :|f+gf+g
(t7) a4.>a2a<  agf+gf+edc+
(t7) |:12 d :|  c+dc+d>  |:8 b :| b4<f+4>bb<c+d |:12 e :|f+gf+g
(t7) >|:8 a :|  a4b4<c+4e4
(t7) |:12 d :|  c+dc+d>  |:8 b :|  bbbbbb <c+d  |:12 e :|ee f+g
(t7) >aaaa aaaa  a4b4<c+4e4
(t7) |:7 d :|>a<  |:5 d :|>a<d>a  b4b4b4b4  b4<c+4d4f+4 |:14 e :|f+g
(t7) |:8 a :|  a4g4f+4e4
(t7) |:8 d :|      |:8 d :|>       bbbbbbbb  b4f+4bb<c+d |:12 e :|f+gf+g
(t7) >a1        a4b4<c+4e4
(t7) |:8 d :|  d4d4c+dc+d>  b4<f+4>b4a4  b4<f+4>b4<c+d
(t7) |:14 e :|f+g  a4.>a2&a<  agf+gf+edc+
(t7) |:7 d :|>a<  ddd>a<d>a<d>a  b4b4b4rf+  b4<c+4d4f+4
(t7) |:14 e :|f+g  a4.a2a  a4g4f+4e4
(t7) |:12 d :|c+dc+d>  |:8 b :|  b4<c+4d4f+4
(t7) eeeee4b4  eeee eef+g  aaaa>a2  <agf+gf+edc+
(t7) |:12 d :|  c+dc+d>  |:8 b :|  bbbbbb <c+d  |:12 e :|f+gf+g
(t7) >|:8 a :|  a4b4<c+4e4
(t7) |:14 d :|c+d  |:>bbbbb4<f+4:|  |:14 e :|f+g  a4.>a2&a<  a1
(t7) |:14 d :|c+d  |:>bbbbb4<f+4:|  |:14 e :|f+g  a4.>a2&a<  a1
(t7) ddddc+dd4  ddddc+da4  >bbbbbbbb  b4<c+4d4f+4
(t7) eeeee4b4   eeeeeef+g  a4.>a&a2<  a4g4f+4e4   ¥05
(t7) ddddc+dd4  ddddc+da4  >bbbbbbbb  b4<c+4d4f+4
(t7) eeeee4b4   eeeeeef+g  a4.>a&a2<  a4g4f+4e4
(t7) ddddc+dd4  ddddc+da4  >bbbbbbbb  b4<c+4d4f+4
(t7) eeeee4b4   eeeeeef+g  a4.>a&a2<  ¥
/D R U M S////////////////////////////////////////////////
(t10) @u127 @v110o2L4  r2|:10r1:|    T115
(t10) |:3 cdcd :|  cdc8d8d8d16d16  c8c8dcd  cdcd cdcd
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) cdcd  cdc8d8d  cdcd  cdc8d8d16d8.  cdcd  cdc8d8d cdcd
(t10) L16 c8d8 'cd' d8. 'c8.d' d 'c8d' dd L4
(t10) cdcd  cdcd  cdcd  cdcd8d16d16  cdcd  cdcd  cdcd  c 'cd' c8d8 'c8d' d16d16
(t10) |:3 cdcd :|
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) cdcd  cdc8d8d  cdcd  cdc8d8r16d8.
(t10) 'c8d'c8dcd  cdcd  cdcd
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) cdcd  cdc8d8d  cdcd  cdc8d8r16d8.
(t10) cdcd  cdc8d8d  cdcd
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) |:3 cdcd :|  cdc8d8d8d16d16
(t10) cdcd  cdc8d8d  cdcd  cdc8d8d16d8.  cdcd  cdc8d8d cdcd
(t10) L16 c8d8 'cd' d8. 'c8.d' d 'c8d' dd L4
(t10) cdcd  cdcd  cdcd  cdcd8d16d16  cdcd  cdcd  cdcd  c 'cd' c8d8 'c8d' d16d16
(t10) |:3 cdcd :|
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) cdcd  cdc8d8d  cdcd  cdc8d8r16d8.
(t10) 'c8d'c8dcd  cdcd  cdcd
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) cdcd  cdc8d8d  cdcd  cdc8d8r16d8.
(t10) cdcd  cdc8d8d  cdcd
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) |:3 cdcd :|  cdc8d8r16d8.
(t10) cdcd  cdc8d8d  cdcd  c8d8 'cd' c 'c8d' d16d16  |:3 cdcd :|  cdcd8d16d16
(t10) |:3 cdcd :|  c L16'cd'dd8 'c8.d'd 'c8d' dd L4  |:3 cdcd :|
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) cdcd  cdc8d8d  cdcd  cdc8d8r16d8.
```

LAST CHRISTMAS
by George Michael
(C) 1984 by MORRISON LEAHY MUSIC LTD.
All rights reserved　Used by permission
Rights for Japan administered by
WARNER/CHAPPELL MUSIC,JAPAN K.K.,c/o NICHION,INC.

```
(t10) cdcd  cdc8d8d  cdcd
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) |:3 cdcd :|  cdcd8d16d16  |:3 cdcd :|  cdc8d8d8d16d16
(t10) |:3 cdcd :|
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) cdcd  cdc8d8d  cdcd  cdc8d8r16d8.
(t10) cdcd  cdc8d8d  cdcd
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) |:3 cdcd :|  cdc8d8d8d16d16
(t10) |:3 cdcd :|  cd L8cddd16d16  'cd' cL4dcd  cdcd  cd cd
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) |:3 cdcd :|  cd L8cddd16d16  'cd' cL4dcd  cdcd  cd cd
(t10) L16c8d8cd8.'c8.d'd'c8d'ddL4
(t10) |: cdcd  cdcd  cdcd  cd L8cddd16d16 L4 :| ¥06
(t10) |: cdcd  cdcd  cdcd  cd L8cddd16d16 L4 :|
(t10) |: cdcd  cdcd  cdcd  cd L8cddd16d16 L4 :| ¥

(T11) R2|:10 R1 :| @U50L8o2
(t11) |:14 g+ :|  g+16g+16a+  |:12 g+ :|  L16g+g+a+8g+g+g+g+ L8
(t11) |:14 g+ :|  g+16g+16a+  |:8 g+ :|  L16|:4 g+g+a+8 :|  L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8 |:15 g+ :|a+ |:8 g+ :|
(t11) L16|:3 g+g+a+8 :| g+g+a+g+ L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8 |:14 g+ :|g+16g+16a+  |:8 g+ :|
(t11) L16|:3 g+g+a+8 :| g+g+a+g+ L8
(t11) |:14 g+ :|g+16g+16a+  |:8 g+ :|  L16|:3 g+g+a+8 :| g+g+a+g+ L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:14 g+ :|g+16g+16a+  |:8 g+ :|  L16|:3 g+g+a+8 :| g+g+a+g+ L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:15 g+ :|a+         g+16g+16a+|:6 g+ :|
(t11) L16|:3 g+g+a+8 :|  g+g+a+g+ L8
(t11) |:12 g+ :||: g+16g+16a+ :|  |:7 g+ :|g+16g+16  |: a+g+g+g+16g+16 :|
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8 |:15 g+ :|a+ |:8 g+ :|
(t11) L16|:3 g+g+a+8 :| g+g+a+g+ L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8 |:14 g+ :|g+16g+16a+  |:8 g+ :|
(t11) L16|:3 g+g+a+8 :| g+g+a+g+ L8
(t11) |:14 g+ :|g+16g+16a+  |:8 g+ :|  L16|:3 g+g+a+8 :| g+g+a+g+ L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:14 g+ :|g+16g+16a+  |:8 g+ :|  L16|:3 g+g+a+8 :| g+g+a+g+ L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:15 g+ :|a+         g+16g+16a+|:6 g+ :|
(t11) L16|:3 g+g+a+8 :|  g+g+a+g+ L8
(t11) |:12 g+ :||: g+16g+16a+ :|  |:8 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:12 g+ :|  L16g+g+a+8 g+g+a+g+ L8
(t11) |:8 g+ :|  L16|:g+g+a+8 g+g+a+g+:| L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:12 g+ :|  L16g+g+a+8 g+g+a+g+ L8
(t11) |:8 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:15 g+ :|a+  |:8 g+ :|
(t11) L16|:3 g+g+a+8 :|  g+g+a+g+ L8
(t11) |:12 g+ :||: g+16g+16a+ :|  |:7 g+ :|g+16g+16  |: a+g+g+g+16g+16 :|
(t11) |:24 g+ :|
(t11) L16|:3 g+g+a+8 :|  g+g+a+g+ L8
(t11) |:12 g+ :||: g+16g+16a+ :|  |:8 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:24 g+ :|  L16|:4 g+g+a+8 :| L8
(t11) |:15 g+ :|a+  |:8 g+ :|
(t11) L16|:4 g+g+a+8 :| L8
(t11) |:12 g+ :||: g+16g+16a+ :|  |:7 g+ :|g+16g+16  |: a+g+g+g+16g+16 :|
(t11) |:14 g+ :|g+16g+16a+  |:12 g+ :||: g+16g+16a+ :|
(t11) |:14 g+ :|g+16g+16a+  |:8 g+ :|
(t11) L16|:4 g+g+a+8 :| L8
(t11) |:14 g+ :|g+16g+16a+  |:12 g+ :||: g+16g+16a+ :|
(t11) |:14 g+ :|g+16g+16a+  |:8 g+ :|
(t11) L16|:4 g+g+a+8 :| L8
(t11) |:12 g+ :||: g+16g+16a+ :|  |:7 g+ :|g+16g+16  |: a+g+g+g+16g+16 :|
(t11) |:15 g+ :|g+16g+16  |:12 g+ :| |: g+16g+16g+ :|
(t11) |:12 g+ :||: g+16g+16a+ :|  |:7 g+ :|g+16g+16  |: a+g+g+g+16g+16 :|
(t11) |:15 g+ :|g+16g+16  |:12 g+ :| |: g+16g+16g+ :|
(t11) |:12 g+ :||: g+16g+16a+ :|  |:7 g+ :|g+16g+16  |: a+g+g+g+16g+16 :|
(t11) |:15 g+ :|g+16g+16  |:12 g+ :| |: g+16g+16g+ :|

(t12) r2|:10 r1 :| o3@u50L4
(t12) c+r2.  r1  |:3 rc+r2  r1 :|
(t12) |: rc+r2 r1 :|  r1r1  rc+r2 r1
(t12) |: rc+r2 r1 :|  r1r1  rc+r2 r1
(t12) |:4 rc+r2 r1 :|
(t12) |:4 rc+r2 r1 :|
(t12) r1r1  rc+r2 r1  c+r2. r1  r1r1
(t12) |: rc+r2 r1 :|  r1r1  rc+r2 r1
(t12) |: rc+r2 r1 :|  r1r1  rc+r2 r1
(t12) |:4 rc+r2 r1 :|
(t12) |:4 rc+r2 r1 :|
(t12) r1r1  rc+r2 r1  c+r2. r1  rc+r2 r1
(t12) |: r1r1  rc+r2r1  r1r1  r1r1 :|
(t12) |:4 rc+r2 r1 :|
(t12) r1r1  r1r2.c+  c+r2.r1  rc+r2r1
(t12) |:4 rc+r2 r1 :|
(t12) r1r1  rc+r2r1  r1r1  r1r1
(t12) r1r1  |: rc+r2r1 :|  r1r1
(t12) r1r1  |: rc+r2r1 :|  r1r1
(t12) |:6 r1 :| rc+r2r1
(t12) |:6 r1 :| rc+r2r1
(t12) |:6 r1 :| rc+r2r1

/SYNTH4/////////////////////////////////////////

(T4) |:10 R1 :| @17R4 @V40@U100R4 @P64Q6o3 L8
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|  @u80@V45
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) >|:32 'F+A' :||:32 'GB'  :|< / オクターブ注意
(T4) |:32 'F+A' :||:32 'GB'  :|  @U100@V40
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|  @u80@V45
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) >|:32 'F+A' :||:32 'GB'  :|< / オクターブ注意
(T4) |:32 'F+A' :||:32 'GB'  :|  @U100@V40
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:32 'GB'  :|  ¥05
(T4) |:32 'F+A' :||:32 'GB'  :|
(T4) |:32 'F+A' :||:30 'GB'  :|

(T14) |:10 R1 :| o4L8Q6@u120 R2
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
(T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
(T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
(T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
(T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
(T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
(T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
(T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
(T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
(T14) |:4 'DF+' 'C+E' :| @U90 /←注意
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
(T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
(T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
(T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|  >  /←注意
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
(T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
(T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
(T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':| <  /←注意
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
(T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
(T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
(T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|@u120 /←注意
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
(T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
(T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
(T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
(T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
(T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
(T14) |:4 'DF+' 'C+E' :| @U90 /←注意
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
(T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
(T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
(T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
(T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
```

```
   330: (T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|  >  /← 注意
   331: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   332: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   333: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   334: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   335: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   336: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   337: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   338: (T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':| <  /← 注意
   339: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   340: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   341: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   342: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   343: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   344: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   345: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   346: (T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|@u120 /← 注意
   347: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   348: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   349: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   350: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   351: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   352: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   353: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   354: (T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|
   355: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   356: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   357: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   358: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   359: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   360: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   361: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   362: (T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|
   363: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   364: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   365: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   366: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   367: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   368: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   369: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   370: (T14) |:'D16F+':||:'DF+':||:'C+16E':||:'C+E':||:'D16F+':
|'C+E' / 注意
   371: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   372: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   373: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   374: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   375: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   376: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   377: (T14) 'DF+' r |:'DF+':| 'C+E' r 'C+E' '>B<E' / 注意
   378: (T14) |:'D16F+':||:'DF+':||:'C+16E':|'C+E''>B<D'|:'D16F+
':|'C+E' / 注意
   379: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   380: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   381: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   382: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   383: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   384: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   385: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   386: (T14) |:3 'DF+' :|'C+E' r 'DF+' 'C+E' r / 注意
   387: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   388: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   389: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   390: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   391: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   392: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   393: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   394: (T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|
   395: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   396: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   397: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   398: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   399: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   400: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   401: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   402: (T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|
   403: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   404: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   405: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   406: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   407: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   408: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   409: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   410: (T14) |:3 'DF+' :|'C+E' r 'C+E' |:'DF+':|
   411: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   412: (T14) |:3 'C+E' :|'D>B<' r >F+16A16B<D
   413: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   414: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   415: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   416: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   417: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   418: (T14) |:'D16F+':||:'DF+':||:'C+16E':||:'C+E':||:'D16F+':
|'C+E' / 注意
   419: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   420: (T14) |:3 'C+E' :|'D>B<' r >F+16A16B<D
   421: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   422: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   423: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   424: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   425: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   426: (T14) |:'D16F+':||:'DF+':||:'C+16E':||:'C+E':||:'D16F+':
|'C+E'
   427: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   428: (T14) |:3 'C+E' :|'D>B<' r >F+16A16B<D
   429: (T14) 'C+E' r |:'C+E':| 'D>B<' r |:'D>B<':|
   430: (T14) |:3 'C+E' :|'D>B<' r 'C+E''DF+' r
   431: (T14) 'F+A' r |:'F+A':| 'EG' r |:'EG':|
   432: (T14) |:3 'F+A' :|'EG' r 'F+A''GB' r
   433: (T14) 'DF+' r |:'DF+':| 'C+E' r |:'C+E':|
   434: (T14) |:'D16F+':||:'DF+':||:'C+16E':||:'C+E':|
   435:
   436: /エレピ//////////////////////////////////////////////
   437:
   438: (T5) |:18 R1 :| @15@U80r8 @P70o4r8 @v110L8q8r4
   439: (T5) |:8 R1 :| 'A1>A<'r1  'B1>B<'r1  'D1<D>'r1  'D2<d>''
C+2<C+>'  '>B4.<B''>A2<A'r
   440: (T5) A2<A4.>A&  A2AB<C+D  E4D4>B2&  B2<AF+D>B  G1&  G2<B
GEC+>  A1&  A2>GF+E4
   441: (T5) D1<|:15 R1 :|
   442: (T5) |:8 R1 :| 'A1>A<'r1  'B1>B<'r1  'D1<D>'r1  'D2<d>''
C+2<C+>'  '>B4.<B''>A2<A'r
   443: (T5) A2<A4.>A&  A2AB<C+D  E4D4>B2&  B2<AF+D>B  G1&  G2<B
GEC+>  A1&  A2>GF+E4
   444: (T5) D1&  D1   B1& B1  <D1& D1  >A1& A1<
   445: (T5) D1&  D1  >B1& B1  <D1& D1  >A1& A1<
   446: (T5) |:8 R1 :| 'A1>A<'r1  'B1>B<'r1  'D1<D>'r1  'D2<d>''
C+2<C+>'  '>B4.<B''>A2<A'r
   447: (T5) A2<A4.>A&  A2AB<C+D  E4D4>B2&  B2<AF+D>B  'G1DE'& '
G2DE' <BGEC+>
   448: (T5) @D1'D2EA'C+2  >B4.A4<GF+G@D0
   449: (T5) A2<A4.>A&  AGF+GAB<C+D  E4D4>B2&  B2<AF+D>B  'G1DE'
& 'G2DE'  <BGEC+>
   450: (T5) 'D4.EA'A&A2&  AGF+GF+EDC+
   451: (T5) D1&  D2C+6D6E6  C+4.D&D2&  D2C+6D6E6  C+4.D2E&  E1
 'C+4.E''D2F+''EG'&  'E1G'
   452: (T5) |: 'D1F+A'r1 :|  r1r1  r1r1
   453:
   454: /V O C A L////////////////////////////////////////////
   455:
   456: (T3) |:10 R1 :| R2  @08r4 @V097@U100r4 o4q8L8r4 @p64r4@K
0
   457: (T2) |:10 R1 :| R2  @08r4 @V080@U100r4 o4q7L8r4 @p70r4@K
2
   458: (t3) |:7 r1 :|
   459: (T2) q8 r2.rf+&  f+4ab2<d&  d4.f+ed4.&  d1  r2b4.a&  a1&
  A4  R2.>
   460: (t3) q7<e4.e4d4>a
   461: (T2) q7r1@V88
   462: (T3) <eef#d4.>bb  <EEF+4D4.C+  C+DC+>B2R  <F+4.E2>B
   463: (T2) R1 R1  R1  R1
   464: (T3) <f#gf#e2d   C+DC+C+4D4>B&  BA2R4.  <E4.E4D4>A
   465: (T2) r1  R1  R1  <C+4.C+4>B4F+
   466: (T3) <eef#d4.>bb     <EEF+4D4.C+   C+DC+>B2R  <F+4.E2>B
   467: (T2) <c#c#d>b4.f#f#  <C+C+D4>B4.A  ABBF+<C+DC+>B&  B2.RF
+
   468: (T3) <f#gf#e2d    C+DC+C+4D4>B&  BA2R4.
   469: (T2) <c#edc#2>R8  R1              R4.<C+4>A4.
   470: (T3) |:8 R1 :|
   471: (T2) |:8 R1 :|
   472: (T3) r4<e4ed4d  A4.F+&F+32&E16.&D4.  Reeeed4d  bb4f+4d4d
>
   473: (T2) r1  R1  r1  r1
   474: (T3) r4<f#&>b16&<d.e4.  f+f+d>B16<D.E4.  RDC+4C+4RC+  DC
+.>B16A4.R4
   475: (T2) r1  R1  R1  R1
   476: (T3) r4r4r4r<d  C+DC+DC+D4.  E4DC+4D4C+&  C+DC+DC+4D4>
   477: (T2) R1  R1  R1  R1
   478: (T3) <f#ede4edf#&  F+EDE4DD4  C+RC+C+RC+DC+  DC+D>A2R
   479: (T2) r1  R1  R1  R1
   480: (T3) <e4.e4d4>a8  <EEF+D4.>BB   <EEF+4D4.C+  C+DC+>B2R
   481: (T2) r1  R1  R1   R4.B<C+DC+>B&
   482: (T3) <f#4.e2>b  <F+GF+E2D  C+DC+C+4D4>B&  BA2R4.
   483: (T2) b1 R1  R1  R4.<C+4>A4.
   484: (T3) <e4.e4d4>a8    <EEF+D4.>BB       <EEF+4D4.C+    C+DC
+>B2R
   485: (T2) <c#4.c#4>b4f#  <C+C+C+>B4.F+F+  <C+C+D4>B4.F+  <GF+
EF+C+DC+>B&
   486: (T3) <f#4.e2>b  <F+GF+E2D  C+DC+C+4D4>B&  BA2. <E>
   487: (T2) b2R2 <dddc#2>R  R1  r4.<c#4>a4.
   488: (T3) <d1  r2edc#c#&  c#dd2.> R1
   489: (T2) r1 R1  R1  R1
   490: (T3) R1  R1  R1 R1
   491: (T2) R1  R1  q8 R2.@V70  R<<D  C+4.>B4A4.&
   492: (T3) r4<e4.d8d4  RC+DA4ED4  R>B<E4.DC+D  RC+DB4 q8F+ED>
q7
   493: (T2) A4>R2.@V90  R1  R1     R1
   494: (T3) r<f#f#>b<dde4  F+F+d>b16<d.e4.  f#4rddc#c#c#  c#>bb
aarar
   495: (T2) r1  R1  R1  R1
   496: (T3) <e4dc#c#d>aa <edc#d2>a <e4ddc#d>aa <edc#d2&d>
   497: (T2) R1 R1 R1 q8   <a4.f#4.f#.e16&
   498: (T3) R1 <F+4.E4.ED  F+EEDDC+DC+  DC+D>A2R
   499: (T2) e4&d2.q8> R1 R1 R1
   500: (T3) <e4.e4d4>a8   <EEF+D4.>BB  <E E F+4D4.C+  C+DC+>B2R
   501: (T2) r1            r1           <C+C+D4>B4.R   r4.b<c#dc
#>b&
   502: (T3) <f#4.e2>b  <F+GF+E2D  C+DC+C+4D4>B&  BA2R4.
   503: (T2) b1          R1          R1             r4.<c#4>a4.
   504: (T3) <e4.e4d4>a    <A6F+6E6&ED>BB  <EEF+4D4.C+  C+DC+>B2
R
   505: (T2) <c#4.a4f#4>R  <DDF+D4.R4>     r2.q8<ab>    <a2g4f#g
>
   506: (T3) <f#4.e2>b  <F+GF+E2D  C+DC+C+4D4>B&  BA2R4.
   507: (T2) q8<f#ee2.>  R1  R1    r4.q5<d q8 ba4e>
   508: (T3) <E4DDC+C+>AA  <EDC+D2>A      <E4DDC+D>AA  <EDC+D2&D>
   509: (T2) q8<d1            r4r4f#6g6a6&  a1&          a1&>
   510: (T3) R1  R2R4.<D     C+DC+C+4D4.  R2R4.C+&
   511: (T2) <a4.q8g4.f#4&  f#4e2>R4      r4r4r4.<d  c#dc#c#4d4>R
   512: (T3) c#>a4.r<f#4d&   d2r2  rgf#2.&  f#1>
   513: (T2) R1                R1    R1       R1
   514: (T3) r1  R2R4.<D   C+DC+C+4D4D   C+DC+C+4D4C+&
   515: (T2) r1  R<ED4>R2  R4.R2<F+&      F+EEF+4G4F+&
   516: (T3) c#>a4.R2             R2R4.<D     C+DC+C+4D4D  C+DC+D
```

```
4ED4>
(T2) f#q8r4r4gf+g16a16&  AGF+ED4>R4  R2R<GF+A&    agf#ed
4>R4
(T3) R1  R2R4.<D  C+DC+C+4D4D  R1
(T2) r1  R1        R1           <C+DC+C+4D4C+&
(T3) R  >a2R4.     R1           R1        R1
(T2) C+rr4rf#ga&  agf#e16d16&d2  r2rf#ga&  agf#e16d16&d2
>
(T3) r1  r2r4.<d  c#dc# q8 c#4d4.  F+2G4.F+&
(T2) r1  R1        R1           R1
(T3) f+1&  F+1          R1          R1  >
(T2) r1     r2<f+6g6a6&  a4.f+ q8 ed4.&  d1>
(T3) r1 r2r4.<d  c#dc#c#4d4d  R1       >
(T2) r1 R1        R1           <C+DC+C+4D4.>
(T3) <e4.f#4d4>a  <a6f#6e6e4d4&  D1>    R1
(T2) r1              R1           <R2.AB  A2G4F+G>
(T3) R1          r2r4.<d   c#dc#c#4d4d  R1          >   @14
@V80
(T2) <F+E2.>R  R1         R1             <C+DC+C+4D4.>   @14
@V70 R16
(T3) q8<e4.d2>a  <eef#4d4.>a  <e4f#4d4.>b  <c#dc#>b2R
(T2) q8<e4.d2>a  <eef#4d4.>a  <e4f#4d4.>b  <c#dc#>b2R
(T3) q8<f#4.e2>b  <f#gf#e2d  c#dc#c#4d4c#&  c#>a2.R  ¥06
(T2) q8<f#4.e2>b  <f#gf#e2d  c#dc#c#4d4c#&  c#>a2.R  ¥06
(T3) q8<e4edr4.>a  <eef#4d4.>a  <e4f#4d4r>b  <c#dc#>b4.R
4
(T2) q8<e4edr4.>a  <eef#4d4.>a  <e4f#4d4r>b  <c#dc#>b4.R
4
(T3) q8<f#4.e4.r>b  <f#gf#e2d  c#dc#c#4d4c#&  c#>a2.R
(T2) q8<f#4.e4.r>b  <f#gf#e2d  c#dc#c#4d4c#&  c#>a2.R
(T3) q8<e4edr4.>a  <eef#4d4.>a  <e4f#4d4r>b  <c#dc#>b4.R
4
(T2) q8<e4edr4.>a  <eef#4d4.>a  <e4f#4d4r>b  <c#dc#>b4.R
4
(T3) q8<f#4.e4.r>b  <f#gf#e2d                 R1 ¥
(T2) q8<f#4.e4.r>b  <f#gf#e2d                 R1 ¥

/鈴 ///////////////////////////////////////////////////////

(t16) @1@K-5 o6 r2 L8v15  q5p1 |:10 r1 :|
(t16) |:24 r1 :|
(t16) |:8 GGGG GGGG :|
(t16) |:2 GGGG GGGG :| ¥15 GGGG GGGG ¥ |:5 R1 :|
(t16) |:8 R1 :|
(t16) |:16 r1 :|
(t16) |:8 GGGG GGGG :|
(t16) |:2 GGGG GGGG :| ¥15 GGGG GGGG ¥ |:5 R1 :|
(t16) |:8 R1 :|
(t16) |:24 r1 :|
(t16) |:40 GGGG GGGG :|
(t16) |:8 GGGG GGGG :| ¥15
(t16) |:8 GGGG GGGG :|
(t16) |:8 GGGG GGGG :| ¥

(t17) @1@K-4 o6 r2 L8v15  q5p2 |:10 r1 :|
(t17) |:24 r1 :|
(t17) |:8 GGGG GGGG :|
(t17) |:2 GGGG GGGG :| ¥15 GGGG GGGG ¥ |:5 R1 :|
(t17) |:8 R1 :| V15
(t17) |:16 r1 :|
(t17) |:8 GGGG GGGG :|
(t17) |:2 GGGG GGGG :| ¥15 GGGG GGGG ¥ |:5 R1 :|
(t17) |:8 R1 :| V15
(t17) |:24 r1 :|
(t17) |:40 GGGG GGGG :|
(t17) |:8 GGGG GGGG :| ¥15
(t17) |:8 GGGG GGGG :|
(t17) |:8 GGGG GGGG :| ¥

(p)    / Merry Christmas to You & Your Lovers.
```

リスト2 LAST CHRISTMAS用カウンタ表示

```
 1:00000000 00000000   2:00008B2C 00000000   3:00008B20 00000000   4:00008BB0 00000000
 5:00006DE0 00000000   7:00008B20 00000000   9:0000078C 00000000  20:0000078C 00000000
21:0000078C 00000000  10:00008BE0 00000000  11:00008BE0 00000000  12:00008BE0 00000000
14:00008BB0 00000000  16:00008BE0 00000000  17:00008BE0 00000000
```

リスト3 闇の血族

Copyright 1990 SYSTEM SACOM

```
  10 /*
  20 /*              "次回予告のテーマ"
  30 /*
  40 /*
  50 /*        闇の血族 X68000      (C) SYSTEM SACOM
  60 /*
  70 /*              arranged & programed by   M.ONO
  80 /*
  90 /*
 100 dim char synthbrass(4,10)={                   /* synthbrass
 110 /*  AF  OM  WF  SY  SP PMD AMD PMS AMD PAN
 120     58, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 130 /*  AR D1R D2R  RR D1L  TL  RS  ML DT1 DT2 AME
 140     31,  3,  4,  5,  3, 25,  0,  1,  1,  0,  0,
 150     30,  5,  8,  6,  5, 30,  0,  2,  1,  0,  0,
 160     28,  7,  5,  6,  3, 40,  0,  1,  5,  0,  0,
 170     30,  2,  5,  7,  5,  0,  0,  1,  3,  0,  0}
 180 m_vset(70,synthbrass)
 190 /*
 200 dim char readsynth(4,10)={                    /* readsynth
 210 /*  AF  OM  WF  SY  SP PMD AMD PMS AMD PAN
 220     58, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 230 /*  AR D1R D2R  RR D1L  TL  RS  ML DT1 DT2 AME
 240     30,  4,  8,  5,  3, 25,  0,  1,  1,  0,  0,
 250     25,  5,  8,  6,  4, 30,  0,  2,  1,  0,  0,
 260     20,  5,  5,  6,  3, 40,  0,  1,  5,  0,  0,
 270     20,  2,  5,  7,  5,  0,  0,  1,  3,  0,  0}
 280 m_vset(71,readsynth)
 290 /*
 300 dim char dguiter(4,10)={                      /* d.guiter
 310 /*  AF  OM  WF  SY  SP PMD AMD PMS AMD PAN
 320     40, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 330 /*  AR D1R D2R  RR D1L  TL  RS  ML DT1 DT2 AME
 340     31,  4,  1,  0,  1, 10,  0,  2,  4,  0,  0,
 350     18,  1,  1,  8,  1, 28,  0,  5,  4,  0,  0,
 360     31,  4,  1,  9,  1, 23,  0,  7,  7,  0,  0,
 370     31, 12,  2,  8,  1,  0,  0,  1,  7,  0,  0}
 380 m_vset(72,dguiter)
 390 /*
 400 dim char synthvoice(4,10)={                   /* synthvoice
 410 /*  AF  OM  WF  SY  SP PMD AMD PMS AMD PAN
 420     58, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 430 /*  AR D1R D2R  RR D1L  TL  RS  ML DT1 DT2 AME
 440     25, 11,  0,  3,  1, 37,  0,  1,  4,  0,  0,
 450     28, 12, 12, 12,  5, 37,  0,  9,  4,  0,  0,
 460     25, 16,  0, 11,  1, 47,  0,  2,  4,  0,  0,
 470     17, 10,  0, 10,  1,  0,  0,  4,  4,  0,  0}
 480 m_vset(73,synthvoice)
 490 /*
 500 dim char chopperbass(4,10)={                  /* chopperbass
 510 /*  AF  OM  WF  SY  SP PMD AMD PMS AMD PAN
 520     32, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 530 /*  AR D1R D2R  RR D1L  TL  RS  ML DT1 DT2 AME
 540     31,  7,  7,  9,  2, 25,  3,  6,  3,  0,  0,
 550     31,  6,  6,  9,  1, 55,  3,  5,  3,  0,  0,
 560     31,  9,  6,  9,  1, 19,  2,  0,  3,  0,  0,
 570     31,  6,  8,  9, 15,  0,  2,  1,  3,  0,  0}
 580 m_vset(74,chopperbass)
 590 /*
 600 dim char hihat(4,10)={                        /* hihat
 610 /*  AF  OM  WF  SY  SP PMD AMD PMS AMD PAN
 620     27, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 630 /*  AR D1R D2R  RR D1L  TL  RS  ML DT1 DT2 AME
 640     31,  0,  0, 15, 15, 41,  0,  6,  0,  0,  0,
 650     31,  7,  8,  7,  8,  7,  0,  5,  0,  3,  0,
 660     31, 20,  0, 15, 15,  7,  0,  3,  0,  0,  0,
 670     31, 18,  0, 10, 15,  0,  0,  0,  0,  3,  0}
 680 m_vset(75,hihat)
 690 /*
 700 dim char snaredrum(4,10)={                    /* snaredrum
 710 /*  AF  OM  WF  SY  SP PMD AMD PMS AMD PAN
 720     60, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 730 /*  AR D1R D2R  RR D1L  TL  RS  ML DT1 DT2 AME
 740     31,  0,  0,  0,  0,  5,  0,  2,  0,  0,  0,
 750     31, 14, 12, 12, 14,  0,  0,  0,  0,  0,  0,
 760     31, 25,  0,  0, 15,  0,  0,  1,  0,  1,  0,
 770     31, 16, 16, 10, 15,  0,  0,  1,  0,  0,  0}
 780 m_vset(76,snaredrum)
 790 /*
 800 dim char bassdrum(4,10)={                     /* bassdrum
 810 /*  AF  OM  WF  SY  SP PMD AMD PMS AMD PAN
 820     59, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 830 /*  AR D1R D2R  RR D1L  TL  RS  ML DT1 DT2 AME
 840     31,  0,  0,  0,  0,  0,  0, 15,  0,  3,  0,
 850     27, 22,  7,  4,  5, 23,  1,  0,  0,  0,  0,
 860     31, 26,  5,  9, 14,  0,  2,  1,  0,  0,  0,
 870     31,  9,  8,  7,  4,  0,  1,  0,  0,  0,  0}
 880 m_vset(77,bassdrum)
 890 /*
 900 dim char crush(4,10)={                        /* crush
 910 /*  AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 920     44, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 930 /*  AR D1R D2R  RR D1L  TL  RS  ML DT1 DT2 AME
 940     31,  5,  6,  0,  5,  0,  0, 15,  7,  1,  0,
 950     31, 10,  4,  5,  4,  5,  0,  3,  0,  2,  0,
 960     31, 20,  6,  3,  3,  0,  0,  1,  7,  2,  0,
 970     31, 25,  6,  5,  7,  0,  2,  7,  0,  3,  0}
 980 m_vset(78,crush)
 990 /*
1000 /*
1010 /*              初期設定
1020 /*
1030 m_init():for i=1 to 8:m_alloc(i,5000):next
1040            for i=1 to 8:m_assign(i,i)  :next
```

```
1050 str a(12)[256],b(12)[256],c(10)[256],d(10)[256]
1060 str e(15)[256],f(11)[256],g(11)[256], h(7)[256]
1070 str pl1[256],pl2[256]
1080 str a1[256],a2[256],a3[256],h1[256]
1090 str b1[256],b2[256],b3[256],b4[256]
1100 str e1[256],e2[256],e3[256],e4[256],e5[256]
1110 str e6[256],e7[256],e8[256],e9[256]
1120 str f1[256],f2[256],f3[256],f4[256]
1130 str g1[256],g2[256],g3[256],g4[256]
1140 m_tempo(163)
1150 /*
1160 /*
1170 /*           melody part
1180 /*
1190   pl1="@l2e-&y48,124e-&y48,0e&y48,124e&"
1200   pl2="y48,0f&y48,124f&y48,0g-&y48,124g-&@l8y48,0g&l8g"
1210    a1="@72gr4b-r4a1gfgr4<d>r4a1&a4"
1220    a2="@70v15gggggfrf4fe-rdc>b-<c2>g1&g4"
1230    a3="c2c.d.e-f4e-"
1240  a(0)="@l1r v14 o4 q8 l8 p3 "
1250  a(1)=a1+a1+"<"
1260  a(2)=a2+"&g4.<"+a2+"ga-b-<"
1270  a(3)=a3+"dr>b-rg<"+a3+"dr"+pl1+pl2+">b-<"
1280  a(4)=a3+"drdddd4.e-1&e-2&e-"
1290  a(5)=a(3)
1300  a(6)=a3+"frfgb-b-4.<c1&c2&c>
1310  a(7)="r1r1"
1320  a(8)=a(2)+a(3)+a(4)
1330  a(9)=a(5)+a(6)
1340 a(10)="r1r1v15>"
1350 a(11)=a(1)
1360 a(12)="l16>gggrrrgggrrrgggrrrrr<d2.."
1370 /*
1380    b1="e-r4e-r4f1e-de-r4e-r4f1&f4"
1390    b2="e-e-e-e-e-drd4dcr>v11b-ade-2e-1&e-4"
1400    b3=">a-2a-2b-1<"
1410    b4="c2c.>b-.<cd4c"
1420  b(0)=" @70 @l1r v10 o5 q8 l8 p3 y48,40 "
1430  b(1)=b1+b1
1440  b(2)=b2+"&e-4.<"+b2+"e-fg<v09"
1450  b(3)=b3+b4+"drg4>b-<"
1460  b(4)=b4+">b-rb-b-b-b-4.<c1&c2&c"
1470  b(5)="v10"+b(3)
1480  b(6)=b4+"drde-ff4.g1&g2&g
1490  b(7)="r1r1"
1500  b(8)=b(2)+b(3)+b(4)
1510  b(9)=b(5)+b(6)
1520 b(10)="r1r1"
1530 b(11)=">v11"+b(1)
1540 b(12)="l16v12gggrrrgggrrrgggrrrrr<v14g2.."
1550 /*
1560 /*
1570 /*
1580 /*           backing synth part
1590 /*
1600  c(0)="v13 q8 y50,8 "
1610  c(1)="@71o3p3c1&c1&c1&c1c1&c1&c1&c1"
1620  c(2)="@71v09o4p2e-1f1g1f1e-1f1g1f1"
1630  c(3)="@71v10p2e-1f1a-2a-2b-1a-2a-2b-2b2b4.<c2&c8"
1640  c(4)="@70v13o5p2l16ge-c>a-<c>a-fdfd>b-gb-ge-c"
1650  c(5)="@71v11o4p2e-1f1a-2a-2b-1a-2a-2b-1b-4.<c1&c2&c8
1660  c(6)="@70v14o3p2l16ce-gb-gb-<dfdfa-<c>a-<ce-g"
1670  c(7)="<c>ge-ce-c>a-fa-fd>b-<d>b-ge-"
1680  c(8)=c(2)+c(3)+c(4)+c(5)
1690  c(9)=c(6)+c(7)+"v14"+c(1)
1700 c(10)="@70l16p2o3v10gggrrrgggrrrgggrr4v12<d2..
1710 /*
1720  d(0)="v13 q8 y51,20 @l1r"
1730  d(1)="@71o2p3c1&c1&c1&c1c1&c1&c1&c1"
1740  d(2)="@71v09o3p1y51,8g1a-1b-1a-1g1a-1b-1a-1"
1750  d(3)="@71v10p1a-1b-1<c2c2>b-1<c2c2d2d2d4.e-2&e-8"
1760  d(4)="@70v13o6p1l16ge-c>a-<c>a-fdfd>b-gb-ge-c"
1770  d(5)="@71v11o3p1a-1b-1<c2c2d1c2c2d1d4.e1&e2&e8"
1780  d(6)="@70v14o4p1l16ce-gb-gb-<dfdfa-<c>a-<ce-g"
1790  d(7)="<c>ge-ce-c>a-fa-fd>b-<d>b-ge-"
1800  d(8)=d(2)+d(3)+d(4)+d(5)
1810  d(9)=d(6)+d(7)+"y51,20v14"+d(1)
1820 d(10)="@70l16p1o4v10gggrrrgggrrrgggrr4v12d2..
1830 /*
1840 /*
1850 /*
1860 /*           bass part
1870 /*
1880    e1="gfe-fgfe-fgfe-fgfe-f"
1890    e2="agfgagfgagfgagfg"
1900    e3=">cr<cc>cr<cc"
1910    e4=">fr<ff>fr<ff"
1920    e5=">gr<gg>gr<gg"
1930    e6=">a-r<a-a->a-r<a-a-"
1940    e7=">b-r<b-b->b-r<b-b-"
1950  e(0)="@73 o4 q8 l16 p3"
1960  e(1)="v10"+e1+e2+e1+e2+e1+e2+e1+e2
1970  e(2)=" @74 "
1980  e(3)="@v120"+e3+e3+e4+e4+e3+e3+e4+e4
1990  e(4)=e3+e3+e4+e4+e3+e3+e4+e4
2000  e(5)=">"+e4+e4+e5+e5+e6+e6+e5+e5
2010  e(6)=e6+e6+e7+e5+"<"+e3+e3+e3+e3
2020  e(7)="@v121"+e4+e4+">"+e5+e5+e6+e6+e5+e5
2030  e(8)=e6+e6+e7+e5+"<"+e3+e3+e3+e3
2040  e(9)=e3+e3+e3+e3
2050 e(10)=e(3)+e(4)
2060 e(11)=e(5)+e(6)
2070 e(12)=e(7)+e(8)
2080 e(13)=e(9)
2090 e(14)="@73v11"+e1+e2+e1+e2+e1+e2+e1+e2
2100 e(15)="@72o2v12gggrrrgggrrrgggrr4<<v14g2..
2110 /*
2120 /*
2130 /*
2140 /*           drums part
2150 /*
2160    f1="@78@v127o5p1q8g2@77v14o2p1q2gggr4r"
2170    f2="gr4rgggggr4r"
2180    f3="@78o5@v127p1q8g2@77v14o2p1q2gr8.gr8."
2190    f4="gr8.gr8.gr8.gr8."
2200  f(0)="p1 l16 "
2210  f(1)="r1r1r1r1r1r1r1@l192rl16"
2220  f(2)=f1+f2+f2+f2+f2+f2+f2+f2
2230  f(3)=f(2)
2240  f(4)=f(2)
2250  f(5)=f2+f2
2260  f(6)=f(2)
2270  f(7)=f(2)
2280  f(8)=f(2)
2290  f(9)=f2+f2
2300 f(10)=f3+f4+f4+f4+f3+f4+f4+f4
2310 f(11)="r1r4@78o5q8g2.
2320 /*
2330    g1="b8b8bbb8bbb8b8b8"
2340    g2="bbb8bbb8bbb8b8b8"
2350    g3="bbb8bbb8bbb8bbb8"
2360    g4="bbbbbbbbbbbbbbbb"
2370  g(0)="@75 v14 o4 q8 l16 p2"
2380  g(1)=g1+g2+g1+g3+g1+g2+g1+g3
2390  g(2)=g4+g4+g4+g4+g4+g4+g4+g4
2400  g(3)=g4+g4+g4+g4+g4+g4+g4+g4
2410  g(4)=g4+g4+g4+g4+g4+g4+g4+g4
2420  g(5)=g4+g4
2430  g(6)=g4+g4+g4+g4+g4+g4+g4+g4
2440  g(7)=g4+g4+g4+g4+g4+g4+g4+g4
2450  g(8)=g4+g4+g4+g4+g4+g4+g4+g4
2460  g(9)=g4+g4
2470 g(10)=g1+g2+g1+g3+g1+g2+g1+g3
2480 g(11)="bbb8rrbbb8rrbbb8rrrrb8r2."
2490 /*
2500   h1="r4gr4..gr4..gr4..gr8."
2510 h(0)="@76 o2 q8 l16 p3"
2520 h(1)="r1r1r1r1r1r1v14p1ggggv13p3ffffv13p2eeeev15p3gggg"
2530 h(2)=h1+h1+h1+"r4gr4..gr4..gr4..gggg"
2540 h(3)=h1+h1+h1+"r4gr4..gr4..gr4..v14p1gv15p3fv14p2ev15p3r'
2550 h(4)=h1+h1+h1+h1+"r4gr4..gr4..gr8.gggggggg"
2560 h(5)=h(2)+h(3)
2570 h(6)=h(4)+h(1)
2580 h(7)="fferdrfferdrffeedr8.g2..
2590 /*
2600 /*
2610 /*
2620 for i=0 to 12:m_trk(1,a(i)):next
2630 for i=0 to 12:m_trk(2,b(i)):next
2640 for i=0 to 10:m_trk(3,c(i)):next
2650 for i=0 to 10:m_trk(4,d(i)):next
2660 for i=0 to 15:m_trk(5,e(i)):next
2670 for i=0 to 11:m_trk(6,f(i)):next
2680 for i=0 to 11:m_trk(7,g(i)):next
2690 for i=0 to  7:m_trk(8,h(i)):next
2700 /*
2710                              m_play()
```

リスト4 ユーフォリー

Copyright 1988 SYSTEM SACOM

```
10 '   EUPHORY            >>> OPENNING <<<
20 '
30 '              Composed by 斉藤  学   (C)System Sacom
40 '              Arranged by 西尾  将人 (C)Stone-tone
50 '
60 DEFINT A-Z:RPT=3
70 TEMPO0:CLS0:TEMPO0:PLAY "T160"
80 '
90 '   Melody ( Piano 1 )
100 '
110 A1$=STRING$(7,"R1")
120 A2$="L4 O4 RE-E-E-E-DE-F6G12&G2.A-6C12&C2.R"
130 A3$="RDDE-FDE-F6G12&G1 G6A-12G6F12&FR"
140 A4$="RDDE-FDF6D12F6E-12&E-1R1"
150 A5$="¯ >E-1DE-FE-6E-12&E-1<G2.R>C2&C6C12D6E-12FCE-FE-1D2.R"
160 A6$="E-1E-FE-F6G12&G2.E-6>C12&C2&C6<G12F6E-12E-CDE-F6F12G6F1
2&FR"
170 A7$="GE-FGB-6B-12B-6B-12R2<G1"
180 A8$="¯ G6A-12A-6B-12R6>D-12C6<B-12L6B-A-GB-A-GF2RF12GA-12B-1
"
190 A9$="B->C12D-C12RC12EG12GFE-GFE-DD12DD12RD12E-F12_L2E-E-  E-
E-E-DE-FE-1_"
200 '
210 PLAY "M1 I101 V100 ¥128 P8 Q8 O5 V80 ¯5_5 R1R1R1";:
220 FOR X=1 TO RPT
230 PLAY A1$;:PLAY A2$;:PLAY A3$;:PLAY A2$;:PLAY A4$;:
240 PLAY A5$;:PLAY A6$;:PLAY A7$;:PLAY A8$;:PLAY A9$;:
250 NEXT:PLAY ":";
260 '
270 '   Melody ( Piano 2 )
280 '
290 A6$="L4O5 C1<B>D<B>D6E-12&E-2.<G6>G12&G2&G6E-12D6C12C<A-B->C
```

```
D6D12E-6D12&DR
300 A7$="E-CDE-F6F12F6F12R2R1"
310 A8$="R1R1R1¯<G1_"
320 A9$="L6 GA-12B-E12RE12>CE12E-DCE-DC<B-B-12B-B-12RB-12>CD12_L
2CCCC<BBBBG1"
330 '
340 PLAY "M1 I101 Q7 O5 V95 ¥128 ¯5_5 R1R1R1";:
350 FOR X=1 TO RPT
360 PLAY A1$+STRING$(24,"R1");:
370 PLAY A6$;:PLAY A7$;:PLAY A8$;:PLAY A9$;:
380 NEXT:PLAY ":";
390 '
400 '   Sub Melody ( Piano )
410 '
420 B1$="R1R1E-&E-6E-12&E-2":B2$=STRING$(7,"E-&E-6E-12&E-2")
430 B3$="L4O3 E-&E-6E-12&E-2E-&E-6E-12&E-2<A-&A-6A-12&A-2A-&A-6A
-12&A-A-6A12B-&B-6B-12&B-2B-&B-6B-12&B-2"
440 B4$="A-&A-6A-12&A-2B-6B-12B-6B-12B-6B-12>D"
450 B5$=">E-&E-6E-12&E-E-E-6E-12E-6E-12R6E-12D"
460 B6$="_5C&C6C12&C>C<<B&B6B12&B>B<B-&B-6B-12&B->B-<A&A6A12&A>A
<"
470 B7$="A-&A-6A-12&A->A-<A&A6A12&A>A<B-&B-6B-12&B->B-<B&B6B12&B
>D"
480 B8$="¯5FFFFGGGGA-A-A-A-L12B-6B-B-6B-R6B->D6<B->E-4&E-6E-&E-4
E-4"
490 B9$="L4 C&C6C12&CEF&F6F12&FF<B-&B-6B-12&B->D-"
500 '
510 PLAY "M4 I101 Q7 O3 L4 V85 ¥128 P7"+B1$+"_7";:
520 FOR X=1 TO RPT
530 PLAY B2$;:PLAY B3$;:PLAY B4$;:PLAY B3$;:PLAY B5$;:
540 PLAY B6$;:PLAY B7$;:PLAY B6$;:PLAY B8$;:
550 PLAY B9$+"E-&E-6E-12&E-E-";:PLAY B9$+"<A-1&A-1A-1&A-1>E-&E-6
E-12&E-2";:
560 NEXT:PLAY ":";
570 '
580 '   Strings 1 & Chorus 1
590 '
600 C1$="E-<B-A->":C2$="I125 O5L1 P9_17B-&B->C&CD&DE-R¯17P7"
610 C3$=STRING$(4,"E-C<G>")+STRING$(4,"E-<BG>")
620 C4$=STRING$(4,"E-<B-G>")+STRING$(4,"E-C<A>")
630 C5$=STRING$(4,"E-C<A->")+STRING$(4,"E-C<F>")
640 C6$=STRING$(4,"E-<B-G>")+STRING$(4,"D<BG>")
650 C7$=STRING$(4,"O4FA->C<")+STRING$(4,"GB->D<")+STRING$(4,"A->
CE-<")
660 '
670 PLAY "M3 I120 O5 L12 V90 ¥128 P7"+STRING$(12,C1$);:
680 FOR X=1 TO RPT
690 PLAY STRING$(28,C1$);:PLAY STRING$(8,"R1");:PLAY C2$;:
700 PLAY "I120L12O5_20"+C3$;:PLAY C4$;:PLAY C5$;:PLAY C6$;:PLAY
C3$;:PLAY C4$;:
710 PLAY C7$;:PLAY "O5E-6&E-12E-6E-12R2 _10O4L1E-EFDE-EFDRRRRL12
O5¯30"+STRING$(4,"E-<B-A->");:
720 NEXT:PLAY ":";
730 '
740 '   Strings 2 & Chorus 2
750 '
760 PLAY "M2 I120 O4L1 V90 ¥128 P6 RRG";:
770 FOR X=1 TO RPT
780 PLAY "L1O4FA-GGFA-G RRRRRRRR P4I125 _17O5G&GE-&E-A-&A-GR¯17P
6";:
790 PLAY STRING$(28,"R")+"I120O4G";:
800 NEXT:PLAY ":";
810 '
820 '   Bass Drum & Snare Drum & Tom-Tom
830 '
840 D1$="R1R2O6L12 EEEEEEL4O1BO6EO1BO6E"
850 D2$=STRING$(12,"L4O1BO6E")+"O1BRL12O3CC<AAFF"
860 D3$=STRING$(14,"L4O1BO6E")+"E6E12E6E12R2"
870 D4$=STRING$(14,"L4O1BO6E")+"E6E12E6E12R6O3C12C12<A12F12O6"
880 D5$=STRING$(7,"E&E6E12&ER")+"E&E6E12&E E12E12E12"
890 D6$=STRING$(4,"E&E6E12&ER")+"EEEEEEE12E12E12EEEE E6E12E6E12
R6E12O3C12<A12F12"
900 D7$=STRING$(16,"L4O1BO6E")+"E2R2R2.E12E12E12E2R2R2O3L12CC<AA
FFL4O1BO6EO1BO6E"
910 '
920 PLAY "M10 Q8 L4 V65 ¥128 "+D1$;:
930 FOR X=1 TO RPT
940 PLAY D2$;:PLAY D3$;:PLAY D4$;:
950 PLAY D5$;:PLAY D6$;:PLAY D7$;:
960 NEXT:PLAY ":";
970 '
980 '   Hi-Hat
990 '
1000 E1$="R1R1L12"+STRING$(12,"O2F+")
1010 E2$=STRING$(72,"F+")+"R1"
1020 E3$=STRING$(84,"F+")+"R2R6F+F+F+F+"
1030 E4$=STRING$(84,"F+")+"R1"
1040 E5$=STRING$(7,"R2.F+F+F+")+"R1"
1050 E6$=STRING$(4,"R2.F+F+F+")+STRING$(21,"F+")+"R4"+STRING$(12
,"F+")+"R1"
1060 E7$=STRING$(96,"F+")
1070 E8$="L2RA+A+A+RA+A+R  L12"+STRING$(12,"F+")
1080 '
1090 PLAY "M10 O2 L12 V40 ¥128 Q6"+E1$;:
1100 FOR X=1 TO RPT
1110 PLAY E2$;:PLAY "_5"+E3$+"¯5";:PLAY E4$;:PLAY E5$;:
1120 PLAY E6$;:PLAY E7$;:PLAY E8$;:
1130 NEXT:PLAY ":"
1140 PLAY ""
1150 '
1160 '  斉藤　学さんの御冥福をお祈りします。
1170 '
```

……………………………………（善）のゲームミュージックでバビンチョ……………………………………

ついにナムコ・ワンダーエッグに行ってきた。28人同時プレイのギャラクシアン[3]は想像以上の迫力。ビデオゲームってここまで進歩したのかぁ，としみじみ実感。ゲーム終了時に，撃墜率が高い順に金銀銅の勲章が各プレイヤーの画面に輝くのだが，私の戦績は1回目は銅賞，2回目は金賞だった。しかし，2回目ともエンディングが見れずにゲームオーバー。うぅぅ……ただ撃墜率が高いだけじゃダメなんだよぉ。

あと，どーでもいいことだが，作戦説明のときのBGMがCDよりもテンポが速いような気がした。

●ソルバルウ　VHS：VIVL-91
ビクター音楽産業　4,900円（税込）　発売中

ナムコのポリゴンシステムによる，3次元体感アクションゲーム「ソルバルウ」の映像を完全収録したビデオが発売された。攻略ビデオというよりは，CGアートビデオという趣の構成がなされている。残念ながらサウンド，ビジュアルの面では前作の「スターブレード」のビデオには及んでいない（展開が冗長な印象を受けた）。しかし，「ゼビウス」世代はとりあえずチェックしておきたい1本といったところか。おまけで，1989年の「NCGA」（米国で開催されるCGコンペ）で優秀作品に選出されたCGアニメーション「ゼビウス」（約2分）が収録されている。

お勧め度　7

●F/A　CD：VICL15011
ビクター音楽産業　1,500円（税込）　発売中

ナムコのシューティングゲーム「F/A」のオリジナル・サウンド・アルバムが早くも登場。売り文句に「これはもうゲームミュージックの反乱だ！」とあるが，まさにそのとおり。聴く人が聴いたら，「こんなのは音楽じゃない」とか叫んでしまいそうな，ハウス風味のノンストップミックスの凄まじい曲たちが延々45分にわたて収録されている。作曲はドラゴンスピリットの「めがてん細江氏」だが，彼のいつもの「泣きのシンセ」と「歌えるメロディ」を期待してかかると，とんでもない目にあうかも。

お勧め度　6

●スナッチャー
―ZOOM TRACKS―　CD：KICA7610
キングレコード　2,800円（税込）　11／21発売

PCエンジン用CD-ROM対応ゲームの「スナッチャー」のBGMサウンドトラックアルバム。ゲームCDでは曲にセリフがかぶっていたため，音楽だけを楽しむことができなかった（らしい）。このアルバムでは「サウンド・トラック」の文字どおり，純粋にBGMを楽しむことができる。原曲であるPC-8801SR版「スナッチャー」の曲たちをしっかり踏まえつつグレードの高いアレンジが施されており，文句をいわせない完成度。単純にFM音源から高価な音源に置き換えたのではなく，曲調に合った曲のための音選択アレンジがなされているので，聴いていて嫌味がない。PCエンジンを持っていない私もすっかり気に入ってしまいました。

お勧め度　9

○スナッチャー　―JOINT DISK―
CD：KICA-7607～7609
キングレコード　4,800円（税込）　11／21発売

「スナッチャー　-ZOOM TRACKS-」と同日に発売になる。こちらはMSX版の「スナッチャー」「SDスナッチャー」のオリジナルゲームサウンドトラックアルバム。全曲収録で，なんとCD3枚組の構成。

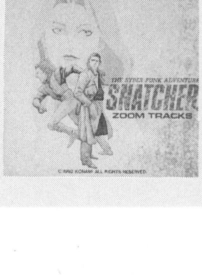
●龍虎の拳　CD：PCCB-00101
ポニーキャニオン　1,500円（税込）　11／20発売

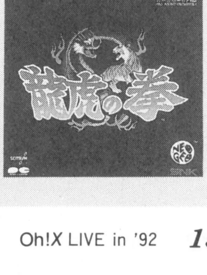
「ストリートファイターII'」を露骨に意識しているこのゲーム，ゲームのほうは本家にちょっと押され気味だけど，ゲームミュージックは本家を超越したパワーがある。大容量をうたっているせいか，PCMサウンドが充実しているのはもちろん，曲も素晴らしいものがそろっている。ハウスミックス調あれば，ジャパニーズ・チャイニーズあり，ハードロックありと，CDアルバムとしても飽きさせない構成となっている。CDにはオリジナルゲームサウンドのほかに，ゲーム中に使用された音声合成が完全収録されていて，これを聴くだけでも楽しいぞ。

お勧め度　8

終わりに

スナッチャーはX68000にも出してもらいたいゲームだなぁ。コナミさん，お願いいい。





(で)のショートプロぱーてぃ――その39

散らかしOK,片づけOK

Komura Satoshi　古村　聡

特別企画に触発されショートプロぱーてぃも派手になったかな？　と思いきや結構マイペースな(で)氏。今月はPICデータをSX-WINDOWのアイコンにする「ICNMAKE.BAS」,ワンタッチでディレクトリ移動ができる「CDS.C」の2本です。

illustration：T.Takahashi

どもども。私が世界一の散らかし男(で)であります。

最近，SX-WINDOWが面白いんですよ。いまさらながら，私もやっとSX-WINDOWをVer.2.01に買い換えたんですけど，このVer.2.01ってアイコンが自由自在に作れるんですよ。知ってはいましたが実際遊んでみるともう楽しくって，楽しくって，これが。

おかげさまで，絵心もないのに魔法使いやら，ミンキーモモやらティンカーベルやら(いっとくが描いているものにぜんぜん脈絡はない)毎日のようにぽこぽこアイコンを作ってしまってるのです。いまではデスクトップが狭い狭い……(そりゃ，作ったアイコン全部デスクトップに並べたうえに，一度開いたウィンドウを開きっぱなしでそこらじゅうに置いてりゃ狭くもなるわな)。やっぱり，日頃から整理整頓を心がけなければいけませんね。

ところで，SX-WINDOWもそうなんですけど，世の中のたいていのウィンドウシステムではディスプレイの全画面のことを「デスクトップ」というんですよね。

しかし，なんで，これがデスクトップなんですかねー，デスクというよりは「壁」っていうほうが合ってるような気がするんですけど。壁紙も貼ってあるし。

はっ，よく見ると，わしの机ってSX-WINDOWのデスクトップ似かもしれない。

表1　使用できるカラーコード

色	カラーコード	R	G	B
黒	&h 0000	0	0	0
白	&h FFFE	31	31	31
明灰	&h B5AC	22	22	22
暗灰	&h 739C	14	14	14
黄	&h 003E	31	31	0
赤	&h 07C0	31	0	0
緑	&h F800	0	31	0
青	&h 003E	0	0	31

(いずれも輝度ビットオフ)

本は開きっぱなしで積み重なってるし，UFOキャッチャーで取ってきたミンキーモモの人形もぶらさがってるし……，それになんといってもこの散らかりよう。うーむ。

そうか，デスクトップってこれを予想して名づけられたのだな。むーん，あなどりがたし，デスクトップ。

明るいSX生活に

さて，お待たせ。今月の1本目のプログラムは茨城県の蓮沼さんの作品で，PICデータをアイコンにしてくれる便利なツール，ICNMAKE.BASです。

ICNMAKE.BAS for X68000
(X-BASIC　要APIC.FNC)
茨城県　蓮沼　勝

このプログラムは，PIC.R用の画像データ，PICデータファイルをSX-WINDOWのアイコンデータにコンバートするためのプログラムで，X-BASICで書かれています（リスト1）。

使用方法ですが，まず，アイコンデータの元になる＊.PICファイルが必要です。自分の使い慣れたグラフィックツールで自由に描くなり，パソコン通信で拾ってくるなりして，データを手元に用意してくださいね。

そして，プログラムを入力して，RUNで実行します。

プログラムを実行すると，まず，変換元のPICデータのファイルネームと作成されるアイコンパターンのファイルネームを聞いてきます。それから，実際に変換する画像の領域を入力します。横ドット数と縦ドット数をコンマで区切って一度に入れてくださいね。

ここまで入れると，プログラムが変換作業を始めます。小さいものなら数十秒，大きなものなら十数分で変換作業が終わります(プログラムをコンパイルしておけば，最大でも十数秒で終了しますよ)。

んー，楽しいっ！　自分のお気に入りのツールでアイコンが描けるってのはいいですよね。やっぱりウィンドウシステムは遊べなきゃウソですよね。そういう意味で考えると，こういうツールの投稿がいままでなかったのが不思議なくらいだったんですけど。いい傾向だなっ。やっぱり，遊べなきゃウィンドウシステムぢゃないよなっ，と思ってしまう(で)なのであります。

ところでこのプログラム，使えるPICデータをアイコンデータに変換する際にちょっとだけ制約があります。

まず，データの画面の使用範囲ですが，画面左上の128×128ドット以内に収めてください。また，ピクセルはデータ変換後縦横比が変わります(変換後は縦長になります)。横のドット数は，16ドット単位でしか設定できません。ですから中途半端なドット数のときには，少し大きめに設定して，パターンエディタに読み込んだあと，不要な部分を削除してセーブし直すといいでしょう。

そして，いちばん大事なのが使用する色です。このプログラムはPICデータで使われている65536色のうち，表1の色以外はすべて透明に変換されてしまいます。データを作るときには十分注意してくださいね。

それと，最後に。このICNMAKE.BASは創刊10周年記念PRO68-Kのディスクに入っているAPIC.FNCが必要です。BASIC.CNFにちゃんと設定してから実行することを忘れないでくださいね。それから，コンパイル後のICNMAKE.Xは単体で動作しますけど，コンパイルするときには当然，C言語用のAPICライブラリ，APICLIB.Lが必要になります。

ちょっと制約の多いのが残念だけど，操作も簡単だし，よくできてますよね。これでSX-WINDOWのアイコンをがんがん作り散らかして，明るいSX Lifeを送ってし

まいましょー(いっとくけど，中森さんのSX版ライフゲームぢゃないぞ)。うーん，ますますデスクトップは散らかるかな，こりゃ。

飛んでディレクトリ

続いて今月の２本目にいきましょう。こちらも散らかす人の強ーい味方。といってもこっちの散らかしはSX-WINDOWではなく，Human68kのコマンドライン派なんですけど。ドライブ中にディレクトリを作り散らかしてしまう人のための便利ツール，大阪府の野崎さんの作品でCDS.Cです。どぞ。

CDS.C for X68000

(要XC ver2.0以上)

大阪府　野崎哲也

このプログラムはリストファイルにディレクトリを登録しておいて，ワンタッチでそのディレクトリに移れるようにするツールです（リスト２)。

まず準備として，このプログラムと同じディレクトリにCDS.LSTというファイル名でディレクトリの一覧表を作ります。書式はドライブ名もつけてフルパスでディレクトリ名を登録をしたいぶんだけ書いておきます。(その見本をリスト３に載せておきますので参考にしてください)。

それから，このプログラムのソースをコンパイルします。例によってエディタでソースリストを入力し，XC ver.2.0以上か，GCCでコンパイルしてください。XCなら，

A> cc /Y cds.c

としてコンパイルします。

で，無事CDS.Xが準備ができたらコマンドライン上で，

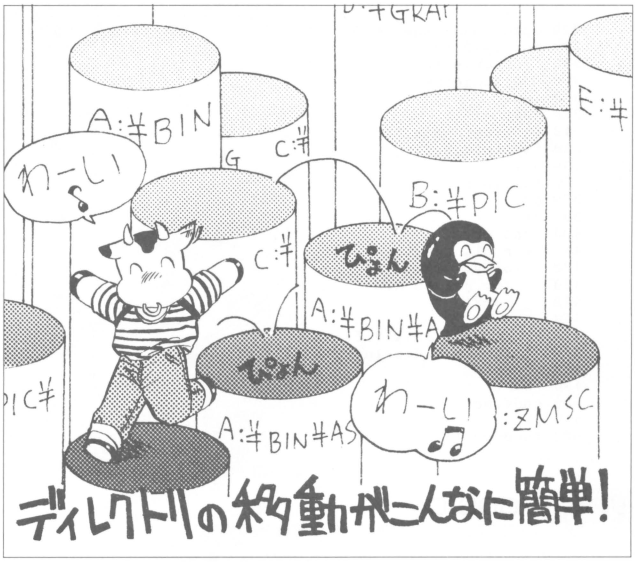
A> CDS

と入力します。すると，画面に登録したディレクトリの名前が番号つきで表示されますので，飛びたいディレクトリの番号を入力してください。すると，登録されていたディレクトリが瞬時にカレントディレクトリになります。また，初めからオプションとして番号も一緒に，

A> CDS 5

という具合に指定すると，さきほどの画面で「5」と押したのと同じディレクトリに，一覧表を画面に出さないでカレントを移します。

うーん，いいですねー。私のようにディレクトリを濫造してしまってわけわからなくしてしまう人には便利なプログラムですね。しかし，こいつで横着してしまうとますますディレクトリをぐちゃぐちゃにしてしまう気もするけど……。要は心がけの問題なのかもしれないな，うん。



あ，そうだ。このプログラムは少し手を加えさせてもらいました。たぶん，エンバグはしてないと思います。

さて，これで，ディスクの整理整頓もできたし，あとは部屋と机の整頓だな。うー，部屋と机を片づけてくれるプログラムがほしい。現実の机の上はデスクトップのようにはいかないのでありました，これが。

そんなわけで，また来月。

動かないよ，と思う前に(2)

今月は，ぱーてぃハンズで掲載した，圧縮してあるキャラクターデータを戻すための注意点を書いていきます。

●ツールは用意できていますか？

今回のキャラクターデータの展開には，ファイルエディタMAC.Xと書庫管理ツールLHA.Xが必要です。MAC.Xは1992年６月号創刊10周年記念PRO-68Kのディスクに，LHA.XはOh! Xすべての付録ディスクについてきています。

まず，MAC.Xでリストを入力します。新規ファイルですから，最初の“New file(y or n)”で“y”を入力します。次に，ファイル名を“PAT2.LZH”などとして“C”(CRC ON)“E”(エディットモード)にして打ち込んでいきます。打ち込む場所は，数字の固まって表示されている部分の両端といちばん下の部分を除いた部分です。そして終了したあと，セーブバイト数3,174でセーブしてください。セーブが終わったら，LHA.Xで展開します。この場合は，

A> LHA X PAT2.LZH

とすればOKです。データを間違いなく打ち込めていれば48Kバイトほどのスプライト定義用のBASICプログラムができるでしょう。

●間違いなく打ち込めてますか？

展開できなかった場合ですが，それは十中八九，打ち間違いです。MAC.Xでデータを打ち込んでいくと，右端といちばん下の部分に２桁の数字が出てきます。これは打ち込んだ数字をチェックするための数字(チェックサムといいます)です。すべて正しく打ち込めた場合，このチェックサムの数値は本に出ている値と同じになります。違っている場合には，もう一度入力部分を調べ直してみましょう。

●CAPSキーは解除してますか？

よくあるミスなのですが，MAC.Xのエディットモードは CAPSキーが押されていると入力を受けつけてくれません。エディットモードに入ってもエディットできない，という場合はこのCAPSキーをよく調べてくださいね。

ではがんばって打ち込んでみてください。

リスト1　ICNMAKE.BAS

```
10 int i,f,plane,v,h,p,px,py,cy,cx,cp,x,y
20 str iname,oname,a
30 print"読み込むPICファイル名を入力して下さい。"
40 print"  拡張子を省略した場合は、'.pic'が付きます。"
50 input iname
60 if instr(1,iname,".")=0 then iname=iname+".pic"
70 print"作成するアイコンファイル名を入力して下さい。"
80 print"  拡張子を省略した場合は、'.icn'が付きます。"
90 input oname
100 if instr(1,oname,".")=0 then oname=oname+".icn"
110 print"X および Y のドット数を入力してください。
120 print"      (プログラムの都合上、X = 16~128 , Y = 1~128 に、"
130 print"                  また、X の値は、強制的に16の倍数になります。)"
140 input x,y
150 if x<16 then x=16
160 if x>128 then x=128
170 if y<1 then y=1
180 if y>128 then y=128
190 cls
200 x=(x¥16)*16
210 dim char pt(3,8)={0,1,0,1,0,1,0,1,0, 0,0,0,1,1,0,0,1,1,
215                   0,0,0,0,0,1,1,1,1, 0,1,1,1,1,1,1,1,1}
220 char hex
230 dim char d(255),lin(16):dim str buf(128)
240 for i=1 to 128:buf(i)=string$(4,chr$(255)):next
250 d={0,0,0,0, 0,0,0,0}:d(5)=x:d(7)=y
260 screen 1,3,1,1:wipe()
270 /*
```

```
280 apic_load(iname,0,0)
290 for cy=0 to y
300  for cx=0 to x
310   cchange(cx,cy)
320  next
330 next
340 f=fopen(oname,"c")
350 fwrite(d,8,f)
360 wr()
370 fclose(f)
380 end
390 /*
400 func wr()
410  for plane=1 to 4
420   for v=1 to y
430    for h=1 to x¥8
440     wr2()
450    next
460    fwrite(lin,x¥8,f)
470    locate 0,0:print y*4;"/";(plane-1)*y+v
480   next
490  next
500 endfunc
510 /*
520 func wr2()
530  pal(plane)
540  lin(h-1)=p
550 endfunc
560 /*
570 func pal(pa;int)
580  p=0:px=(h-1)*8:py=v-1
590  for i=0 to 7
600   p=p*2+pt(pa-1,point(px+i,py))
610  next
620 endfunc
630 /*
640 func cchange(ch;int,cv;int)
650  cp=point(ch,cv)
660  if cp=&HF83E then pset(cx,cy,0):return()
670  if cp=&HB5AC then pset(cx,cy,1):return()
680  if cp=&HFFFE then pset(cx,cy,2):return()
690  if cp=0      then pset(cx,cy,3):return()
700  if cp=&H739C then pset(cx,cy,4):return()
710  if cp=&H7C0  then pset(cx,cy,5):return()
720  if cp=&HFFC0 then pset(cx,cy,6):return()
730  if cp=&H3E   then pset(cx,cy,7):return()
740  if cp=&HF800 then pset(cx,cy,8):return()
750  pset(cx,cy,0)
760 endfunc
```

リスト2 CDS.C

```
/*************************************************
   CALLENT DIRECTORY
            SELECTER    Ver  1.0

                              平成4年9月3日

                        Written By  野崎  哲也
*************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <doslib.h>

FILE *stream;
FILE *stream2;

unsigned char buf[31][68];
unsigned char dummy[1024];
unsigned char *poin;

int i,j,x,y,max=1,no=0;

struct PDBADR *pspadr;

unsigned char *pathGeter( void );

void main( argc , argv )

int argc;
unsigned char *argv[];
{
        unsigned char *path;
        puts("X68k Current Directory Selector v1.0 Copyright 1992 Testsuya.Nozaki");

        if (argc !=1) {
        if ( (*argv[1] == '/') || (*argv[1] == '-') ){
                puts(   "使用法:cds  [番号] ¥n"
                "       番号を指定するとその数字を入力したのと同じ動作をします。");
                exit(1);
                }
                no = atoi(argv[1]);
        }

        path = pathGeter();
        strcpy(dummy,path);
        strcat(dummy,"CDS.LST");
        if ( (stream = fopen(dummy,"r") ) == (FILE *)NULL) {
                puts("      『CDS.LST』が存在しませんので、作業を中断します。");

                exit(1);
        }

        do {
                i = Ffgets(buf[max],68,stream);
                max++;
        } while ( i != (int)NULL );

        fclose(stream);
        max-=2;

        if (max >30) {
                puts("リストの量が多すぎます。");
                exit(1);
        }
        if (no == 0){
                putchar('¥n');

                        for (i=1;i<=max;i++){
                        printf("        %2d = ",i);
                        printf("%-68s¥n",buf[i]);
                }

        printf("¥n              番号を入力して下さい > ");

                no = atoi(gets(dummy));
        }
        if (  (no < 1) || (no >max)  ) {
                printf("¥n      入力された番号が異常なので、作業を終了します。¥n¥n");
                exit(1);
        }
        *dummy      = buf[no][0];
        *(dummy+1) = buf[no][1];
        *(dummy+2) = (unsigned char)NULL;
        system( dummy );

        strcpy( dummy , "CD ");
        strcat( dummy , buf[no] );
        system( dummy );

}
/***********************************************/
int Ffgets(po,le,stre)

unsigned char *po;
int le;
unsigned char *stre;

{
        char *dum;

        dum = fgets(po,le,stre);
        if (dum == NULL) return( 0 );

        while (*po != 0) po++;
        po--;
        *po = 0;
        return( (int)dum );
}
/***********************************************/
unsigned char *pathGeter( void )
{
        pspadr = GETPDB();
        return &(pspadr->exe_path[0]);
}
```

リスト3 CDS.X用サンプルデータ

```
A:¥GAME
A:¥BIN
A:¥SYS
A:¥BASIC
A:¥SHELL
A:¥SMPL
A:¥IPL_BAT
A:¥TEMPORARY
A:¥USER¥C
A:¥USER¥S
A:¥USER¥BIN
A:¥USER¥BASIC
A:¥USER¥WORD
A:¥USER¥CGA
A:¥UTILITY
A:¥UTILITY¥INCLUDE
A:¥UTILITY¥INCLUDE2
A:¥UTILITY¥LIB
A:¥UTILITY¥MUSIC
A:¥アクセサリ
A:¥アクセサリ¥FONT
A:¥アクセサリ¥SAMPLE
```

ぱーてぃハンズ(2)

先月はスプライト6枚を使ってひとり分のデカキャラを表示するところまでやりました。

ところでところで，ゲームといえば，スプライトのほかにも忘れていけないものがあります。そのなかでもいっちばん大事なのが，やっぱりジョイスティックやテンキーなどの入力装置でありましょう。そりゃあ，なんたって，そのテの入力装置がなければキャラクターが動かせない，殴れない，逃げられない，ちっともゲームにならないのですね。って，わけで今月はジョイスティック周りをやってみましょう。

マニュアルを見る

さてさて，それではジョイスティック関係の命令をマニュアルで調べてみましょう。ジョイスティック関係の命令は2つしかありません。それは，

- stick()関数
- strig()関数

です。stick()関数はジョイスティックの状態を調べる関数です。たとえばプログラム中で，

iWork＝stick(1)

とやるとジョイスティック1番(縦型のX68000では前にあるほうのジョイスティック。X68000 PROだと左側だな)の状態がiWorkに入っています。で，スティックが倒されている方向によって図1のような値が入ることになります。0以外はテンキーの位置と同じだから簡単ですよね。

strig()のほうは，

iWork＝strig(1)

とすると同じくジョイスティク1番のトリガボタンの状態が入ります。その内容は，

- 0……ボタンが押されていない
- 1……Aボタンが押された
- 2……Bボタンが押された
- 3……両方押されている

こんな感じです(今回strigは使いません)。

キャラを動かす

で，このジョイスティックの動きに合わせてスプライトを動かします。スプライトの動かし方は簡単です。前回，sp_move()関数でスプライトを表示しましたね。そのsp_move()関数で同じスプライトを違う座標に表示するだけです。たとえば，最初に，

sp_move(1,50,100)

でプレーン1を(50,100)の位置に表示したとします。それから，

sp_move(1,200,100)

とすると(200,100)の位置に動くのです。簡単でしょ。

で，このジョイスティックを使ってキャラクターを動かします。今月はとりあえず，ジョイスティックの動きに合わせて左右に動かしてみましょう。

まず，左に動かす場合です。ジョイスティックが左に倒されているときですから，

if (stick(1)＝4)

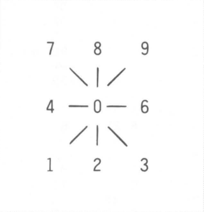
に，元のキャラクターの位置のx座標が左にずれるんだから(16ずれるとして)，

then x＝x－16:

で，スプライト(この場合は0番として)を表示すると，

PutSprite(0,x,y)

図1



以上のようにやればいいわけですよね。逆に右の場合は，

if (stick(1)＝6) then x＝x＋16: PutSprite(0,x,y)

としてやればいいわけです。あとはスプライトが画面からはみださないように画面の右端と左端のチェックをするだけ。てなもんです。

動いたかな?

さて，スプライトは動きましたか？　結構，するすると動くでしょう。

でも，ちょっとうまくないんですよね。そう，キャラクターが足も動かさずに動いてしまうのです。

ということで，来月は足を動かしたり，殴ったり，といったキャラクターのアニメーションができるようにしましょう。いよいよ，ゲームっぽくなってきますよん。

最後に，リスト4にこのぱーてぃハンズで使う，キャラクターデータの圧縮リストを掲載しました。入力方法，解凍方法は本文中の「動かないよ，と思う前に(2)」のコーナーを参考にしてください。

リスト

```
 10 screen 0,0,0,0
 20 int x,y
 30 x=48:y=128
 40 sp_disp(1)
 50 sp_on(0,1)
 60 PutSprite(0,x,y)
 70 while(0=0)
 80 if(stick(1) = 4) then x=x-4:PutSprite(0,x,y)
 90 if(stick(1) = 6) then x=x+4:PutSprite(0,x,y)
100 endwhile
110 end
120 func PutSprite(pnum, x, y)
130    int ix,iy
140    for  ix=0 to 1
150       for iy=0 to 2
160          sp_move(ix+iy*8,x+ix*16,y+iy*16,pnum+ix+iy*8)
170       next
180    next
190 endfunc
```

リスト4 PAT2.LZH

```
0000  20 2A 2D 6C 68 35 2D 43  : F0
0008  0C 00 00 8A BD 00 00 E3  : 36
0010  B2 32 19 20 01 07 70 61  : F6
0018  74 2E 62 61 73 10 28 48  : 58
0020  00 00 08 69 73 9A D2 56  : A6
0028  DB 74 9C 78 04 C9 B8 4B  : 33
0030  0A 89 40 38 1D BB DB BB  : 79
0038  DD D4 0B 8F 7E 01 41 02  : 0D
0040  6E 0B 00 95 C8 1C 73 0A  : 6F
0048  8B F1 B9 35 C7 C7 C7 BC  : 7B
0050  8A 49 17 11 51 19 4A 8C  : 3B
0058  A6 CA 91 2A D2 6D 29 12  : A5
0060  B0 BB 13 11 1C 13 DF F7  : 94
0068  0A 3A E5 64 6D 6C 6C 6C  : 3E
0070  13 B6 3E CF AF CB F4 79  : BD
0078  BC 7F 2F D7 F3 79 BC DE  : 47
------------------------------------
SUM:  C6 94 5D 3F 88 97 13 4B  2F07

0080  3F 2F 93 E4 FE 9F D7 07  : 60
0088  43 E9 FA 7C 7E 6D AF FF  : 3B
0090  C7 E4 F1 7C 7F 12 B9 DB  : 3D
0098  5F 2F E7 FB 7C 9B 3B 18  : DA
00A0  BB FB 7D B3 76 D0 67 D9  : 6C
00A8  F5 FC BB 3F 57 D3 F5 79  : 83
00B0  7E 47 5E 75 E2 9F 61 F7  : 71
00B8  76 0A F2 BC CD 2C EE 25  : 3A
00C0  DD C3 BC C7 67 65 6E EC  : 49
00C8  BE F1 12 E0 ED B5 DD B4  : D4
00D0  BC CA AF C0 25 0B BB 8B  : 6B
00D8  B5 DC 48 60 FB 31 BB B8  : D8
00E0  D7 99 EA B8 3C A6 57 77  : C2
00E8  21 7A CF AA 4E BB B1 1D  : EB
00F0  AF AA 91 6B 3B 85 77 72  : FE
00F8  57 78 61 6E 60 A9 3A F1  : D2
------------------------------------
SUM:  56 02 5D FC 8C 0C 9F 41  55D8
```

```
0100  B9 F0 B0 C0 B1 60 D1 5E  : 59
0108  7B 2B 46 CE EE E6 0D 91  : 2C
0110  DE 45 71 E0 36 DC C1 F2  : 39
0118  DA C2 08 33 F0 7D BD CC  : CD
0120  21 24 BC F4 7B B0 78 AE  : 46
0128  E6 11 12 DE 29 47 07 73  : D1
0130  6B 08 B6 B3 BF CF 9C FF  : 05
0138  1F C4 CE DA FD 8B 3C EF  : 3E
0140  E8 93 6F B9 B7 EF 7E 2F  : F6
0148  A3 FC EC 6C FF 8F 9B CB  : EB
0150  B1 B3 F2 1A AC FB 1B 7E  : B0
0158  EF B3 FD BF D4 FF C2 D8  : CB
0160  F3 FE FE 7F DD D7 BD FC  : DB
0168  CC 2C C0 5B FA C2 DF BD  : 6B
0170  CC C2 D3 9C 2D FC 5C CC  : 4E
0178  2D 40 5B 67 0B C8 06 58  : 60
------------------------------------
SUM:  60 44 F7 DB 6A C5 A7 E9  E781

0180  39 98 5C 3B 42 CB 80 2E  : 23
0188  39 82 F9 01 E1 84 2E 5D  : A5
0190  7E 2F C2 17 B7 B9 2E 50  : 74
0198  BB 3E DE 50 B2 E9 0B 2E  : FB
01A0  30 BE 8D 7B 97 17 33 0B  : E2
01A8  EC D7 BE 40 B4 F0 0B 85  : F5
01B0  F0 B5 1E 2E DD B0 27 B8  : 5D
01B8  2E 61 5E 7D 7B 7F B9 FF  : 1C
01C0  51 9C 4E 3C 5E 3F 9F 6A  : 1D
01C8  5C 5B 33 87 5A 52 1D 68  : A2
01D0  21 D6 96 67 9B FC 02 DF  : 6C
01D8  84 2D FD E1 69 88 2D FA  : A7
01E0  42 D4 05 B0 0B B2 1D 74  : 19
01E8  DF 3F 70 5C 70 85 C2 F8  : 99
01F0  7F 70 5C AA 16 AE 01 64  : 1E
01F8  02 CF 16 3D FB C2 DE 02  : C1
------------------------------------
SUM:  D9 7E B7 07 77 E3 AE CD  C5FE
```

```
0200  D0 05 AE 20 BD 3B 8F 56  : 80
0208  D5 07 39 85 5B 0E B5 C3  : 7B
0210  0E B2 B2 1D 6D 48 75 B0  : 69
0218  87 5B 0A 75 D8 53 AE C2  : FC
0220  9D 76 14 EB B0 A7 5D 85  : 4B
0228  3A EC 29 D7 61 4E BC 32  : C3
0230  53 AE DE EA 75 E0 29 D7  : 1E
0238  86 3D 7B B7 EB 4E BC 05  : EF
0240  3A F0 14 EB C0 53 AF 0A  : F5
0248  D3 AF 0B 61 D7 0C 30 EB  : EC
0250  3B 21 D7 1A 43 AE 22 1D  : 7D
0258  71 14 EB C4 53 AF 11 4E  : 95
0260  BC 45 3A F1 14 EB C4 53  : 42
0268  AF 11 4E BC 45 3A F2 14  : 4F
0270  EB C8 53 AF 21 4E BC 85  : 65
0278  3A F2 14 EB C8 53 AF 2A  : 1F
------------------------------------
SUM:  33 4A 09 0B 3D 89 98 94  2F41

0280  D3 AF 2B 61 D7 2C 30 EB  : 2C
0288  7D F8 75 CD 4E 1B F0 EB  : FB
0290  27 56 1D 64 EA 4B 7D BB  : 6B
0298  C2 F2 CB 7D BA 42 D0 7B  : 43
02A0  1A F7 85 BF 08 5B 6B A8  : CB
02A8  8C 9D 0D 01 6F CE 17 1D  : A8
02B0  35 37 EE 85 E5 A7 5D FF  : C7
02B8  97 3B 61 65 92 9D 77 F5  : 33
02C0  85 90 F1 73 EB AC 1F 20  : 4F
02C8  5B C0 5A 57 31 C9 40 5B  : 61
02D0  6A 0B 98 56 08 75 91 61  : D2
02D8  87 5A 59 0E BA 31 36 41  : AA
02E0  89 B2 0C 4D 97 E8 C4 DE  : B5
02E8  40 B1 0E B3 50 16 C0 2E  : 06
02F0  00 2E 20 2E 55 0B F5 31  : 02
02F8  37 CC 16 35 EE F1 AF 77  : 53
------------------------------------
SUM:  7C 07 F5 4A BF 56 11 96  6780
```

```
0300  8D 7B DC C4 DF EF 4E B9  : 7D
0308  5C C4 DF ED 51 3C C2 DB  : 16
0310  50 5C C2 AD 87 5D 18 9B  : B2
0318  5B 21 D7 46 26 C8 31 36  : EE
0320  48 33 CD 2A 9D 79 25 53  : 00
0328  AF 24 50 17 54 EB C9 2A  : 6C
0330  9D 79 25 53 AF 24 AA 75  : 80
0338  E4 B5 76 1C 96 A9 D7 92  : D3
0340  D5 D8 72 5A A7 5E 4A 9D  : 65
0348  3E 2F F7 A7 5E 4A A3 D8  : 2E
0350  F9 A7 C6 59 79 98 55 B0  : D5
0358  EB A3 13 6D 64 3A E8 C4  : 58
0360  D9 06 26 C9 AA 9D 79 35  : C3
0368  53 AF 26 AA 75 E4 CA 02  : F7
0370  D8 05 D5 3A F2 6A A7 5E  : 4D
0378  4D 54 EB CA 0E 01 64 02  : CB
-----------------------------------
SUM:  54 A0 5A 92 14 E7 40 69  E771

0380  EA 9D 79 41 E0 2D 00 5D  : AB
0388  53 AF 28 6B 4E BC AD 62  : AE
0390  6C A8 C4 DC 2C 87 5D 18  : DC
0398  9B 2F D1 89 BE 18 D7 C4  : 95
03A0  4B 7E 2F D5 1F BB E1 68  : F0
03A8  02 F9 65 BF 8E 50 FB 92  : 8A
03B0  AF 98 5B 69 0B 2C 41 7C  : FF
03B8  94 EB E5 F1 3D D0 B8 F3  : 0D
03C0  85 E3 EB D0 2E 5D 7E 2F  : 5B
03C8  BF E3 FB 61 72 77 78 5E  : BD
03D0  1C C7 91 77 E1 7D 02 CF  : 1A
03D8  9F 31 F3 6D 0E 5D 54 EB  : DA
03E0  F1 EA AE 48 02 EA EC 39  : E2
03E8  55 89 B2 B5 89 B2 A3 13  : 36
03F0  71 BF 0E B3 A3 13 66 EA  : F7
03F8  4B 7F F7 A7 5C DC 60 2F  : 2F
-----------------------------------
SUM:  D5 8C D9 6B 26 C8 57 B0  2EBA

0400  9A 5B E5 9C 2F 96 5B FE  : 94
0408  99 7D CC 2D 78 02 CB B4  : 08
0410  2D 80 5C 33 05 FA E7 B7  : D9
0418  30 B8 F0 85 97 58 5C B8  : 60
0420  C2 FF 3A 1B F6 C2 C9 DC  : 73
0428  14 2D F7 05 97 0E 84 FD  : 63
0430  94 81 CC 2E B4 EB 99 3C  : 83
0438  05 A0 0B 50 17 AE 1D 67  : 49
0440  6B 13 67 46 26 E5 64 3A  : D4
0448  E8 C4 D9 9D 69 D7 33 AD  : 42
0450  3A E6 75 A7 5C CD 00 5F  : C4
0458  75 3A F8 C2 D8 05 C3 4D  : 56
0460  1E 7B A1 79 69 D7 E5 91  : 69
0468  5D 02 F2 53 AF 96 0B FC  : F0
0470  21 7C 4C 4D F6 05 96 60  : 27
0478  B2 E7 0B 3A ED 0D EF 01  : C8
-----------------------------------
SUM:  4F 34 9C BE 59 60 3B 1E  5873

0480  68 02 FB 9B 93 C6 17 58  : C8
0488  75 DA C4 D9 D1 89 B2 75  : 6D
0490  90 EC A3 13 86 85 D3 7C  : 8C
0498  47 DD 39 49 85 9F 3D 0B  : 12
04A0  85 98 2D FA 28 10 E5 0B  : 6C
04A8  4E B0 BD FD CA 4C 2D 6B  : 66
04B0  45 42 36 6A 50 39 E6 58  : EE
04B8  90 CF 51 27 30 5C 40 5C  : FF
04C0  AA 16 AE 01 64 02 CC 05  : A6
04C8  BC 05 A0 0B 50 16 DA AA  : 56
04D0  24 98 55 B0 EC A3 13 84  : E7
04D8  56 43 B2 8C 4E 9F A9 89  : F6
04E0  D2 C4 D6 EE FB 66 B3 0B  : 79
04E8  3A D0 BA CF 01 68 02 D7  : D5
04F0  C0 2E FE 7A 33 0F 17 85  : 44
04F8  6A 25 68 F8 05 9D F0 B9  : 3A
-----------------------------------
SUM:  72 DB 57 CF 03 38 2F 5A  A3AD

0500  77 85 9D A1 70 77 6D 74  : 02
0508  27 77 33 0B 20 17 D5 66  : 4E
0510  89 FA 39 98 5F 45 3B 27  : 5A
0518  98 2D 00 5F 35 3B 27 9C  : 57
0520  2D B5 05 CC 2A D8 76 51  : 7C
0528  89 D2 3B 21 D9 46 27 8E  : 8B
0530  E6 27 BB EF 8A 88 97 0C  : 6C
0538  44 B8 63 AA 5C 29 7C 2F  : 39
0540  E1 2E 14 C2 16 DD 38 F7  : 07
0548  4C EC 4C 41 74 A7 66 CF  : 15
0550  87 EB D4 06 F8 5D 90 EC  : 1D
0558  3B DE EE 7C E0 F1 C2 17  : 2D
0560  2D 21 67 B9 98 F8 18 9E  : B4
0568  FD 81 ED 0B 2E 90 BE 3D  : 2F
0570  2F E0 62 7B 50 5B F9 C2  : 52
0578  F3 6B DF 03 13 DF A8 1E  : F8
-----------------------------------
SUM:  DA 59 1E F0 98 71 BB 3B  FD55

0580  E0 5A F0 85 F4 6D 0D AF  : CC
0588  85 B7 80 5C E9 D9 B5 89  : 18
0590  E3 A3 13 C4 FB F0 EC 7D  : B1
0598  18 9F 7B 8B 5D 02 2F 71  : BC
05A0  80 BA D3 B2 F7 20 0B 5C  : 3D
05A8  E9 67 F3 05 B0 0B 87 4A  : D4
05B0  50 79 42 F1 D3 B3 F9 A7  : 22
05B8  57 B8 17 BB 4E CF B1 35  : E4
05C0  FB 82 C9 C0 2C B8 42 CB  : F7
05C8  AC 2E AB 55 BC B9 35 EF  : 73
05D0  EB 8E E4 C2 D0 05 A8 0B  : A7
05D8  6D 41 73 0A C1 0E C7 D1  : 92
05E0  89 F2 4B 21 D9 46 28 1E  : 4C
05E8  65 AD 96 47 99 EB 4A 0D  : CA
05F0  E7 C8 9D 9F 40 5A 00 BE  : 43
05F8  54 EC F9 82 D8 05 C3 85  : E0
-----------------------------------
SUM:  98 77 5F FD 00 F9 34 AC  1087

0600  57 BF 72 89 6F 38 E7 57  : F6
0608  1F D6 EC 63 CE 59 FC 5D  : C4
0610  DE A7 9C 1E F7 00 B2 D1  : B9
0618  85 FC A1 67 D2 B7 BE 50  : 20
0620  BD DA 76 7D E1 75 A7 65  : EC
0628  EF 50 16 DA 82 E6 15 6C  : 18
0630  3B 2E 62 81 6C 87 65 18  : BC
0638  A0 7A 75 3B 1B C7 1B 14  : DB
0640  3D 68 11 BD 1E 02 D3 5A  : C0
0648  0D EF 4F 82 17 78 73 D1  : A0
0650  39 50 BB F5 DB 2C F4 80  : B4
0658  0B 88 0B 97 68 5F 06 C9  : CB
0660  15 C0 2C 80 59 80 B7 80  : 91
0668  B4 01 7C BA F7 4E 0E 71  : AF
0670  85 EB 76 35 F6 B1 40 FB  : FD
0678  98 A0 6B 21 D9 46 28 1E  : 29
-----------------------------------
SUM:  D4 85 AD DF 87 BB FC 50  B04D

0680  C5 AA F3 F9 FA 23 65 42  : 1F
0688  EF A8 2F 91 0B BE 80 B1  : 51
0690  2E 1D E6 28 3D E1 78 F3  : E2
0698  D3 B8 2E 00 2E 39 C2 CF  : B1
06A0  40 5E 48 76 75 85 F1 31  : 78
06A8  41 D8 16 40 2C F4 05 9E  : 32
06B0  70 B7 80 B4 01 79 21 D9  : CF
06B8  E9 87 06 F8 5D 61 D9 6B  : 70
06C0  14 0F B9 8A 08 59 0E CA  : 9F
06C8  31 40 F8 95 69 41 88 A7  : D7
06D0  66 2F AD 31 B1 14 EC C5  : E9
06D8  74 D1 EB BA 17 A9 0B BB  : 70
06E0  FA 40 8C 3B C2 F0 EA 32  : CF
06E8  39 69 D9 F6 30 10 F8 CB  : 74
06F0  B4 2F 83 64 92 77 3E EF  : 00
06F8  39 02 CB 44 74 3F E4 2A  : 0B
-----------------------------------
SUM:  CE C4 16 F7 A0 5B A0 CF  4F03

0700  26 16 79 E6 01 A2 0A 92  : DA
0708  7E 80 BC F1 84 92 56 73  : 8A
0710  12 DD D7 BA 69 0B 6E 00  : 62
0718  BF 4C 4A 31 C3 B2 D6 28  : F9
0720  1F 73 14 11 BF 0E C4 A3  : EB
0728  14 08 E2 D7 4C 6A 38 F5  : B8
0730  D3 1A 8E 7E BA 63 51 C9  : 30
0738  A1 AE 1E 50 BA 67 A7 C0  : 45
0740  2D B5 EB 61 38 98 A0 EC  : 8A
0748  0B 8E BA 89 53 79 8A 0D  : 3F
0750  39 8F BC C5 06 9D 2F DC  : F7
0758  C5 07 7E AA F0 31 41 2F  : 85
0760  00 B7 F1 85 F4 EE 3F 03  : 51
0768  14 12 FF D0 9C B4 2E 9A  : 0D
0770  F7 F6 53 9B BE 17 AE 9D  : FB
0778  94 B5 8A 04 B9 8A 09 59  : 7C
-----------------------------------
SUM:  F1 4F A4 C5 B8 55 56 E5  A97E

0780  0E CA 31 40 86 5A E8 A9  : BA
0788  43 30 17 56 A5 50 D3 5D  : 05
0790  28 28 6B AE 94 14 36 01  : 48
0798  7D D4 EC F0 6A 30 E3 AE  : 58
07A0  98 D4 3E CA 76 78 76 48  : 20
07A8  F7 00 B2 EB A8 97 20 59  : 4C
07B0  F4 D4 4B C7 00 87 BC 05  : 22
07B8  A7 38 5E 68 50 3D 40 5E  : D0
07C0  3D 7B FB 62 81 30 AB 61  : D2
07C8  D9 46 28 0D D6 43 B2 8C  : AB
07D0  50 27 B5 8A 0E 3F 44 FF  : 46
07D8  AC 50 72 05 BC 05 A0 0B  : DF
07E0  50 16 C0 2E 00 2E 20 2E  : D0
07E8  55 0B F5 31 41 CC 16 35  : DE
07F0  EE A3 5E EA 35 EE A3 5E  : FD
07F8  EA 35 EF 56 28 12 D6 28  : 9C
-----------------------------------
SUM:  AF 07 84 B5 56 72 56 99  399B

0800  12 8C 50 19 59 0E CA 31  : 69
0808  40 8C 55 A5 05 85 3B 2D  : B8
0810  54 EC D1 BA 93 B3 FB A6  : B2
0818  13 30 BE CC F4 C9 E2 ED  : 59
0820  AD F0 94 68 0C C7 8D 6B  : 64
0828  09 6E 48 76 27 0D AC 5C  : 71
0830  01 70 76 87 04 7D 2A 25  : 3E
0838  A4 39 A1 D8 9E 9A AF 61  : 9E
0840  D1 0E CF BE 8B 59 85 88  : 5D
0848  76 40 43 B2 02 1D 95 62  : C1
0850  81 2D 62 81 28 C5 01 9D  : 1C
0858  90 EC A3 14 09 12 AA 59  : 51
0860  F1 3D 01 79 EA EA 8B C0  : C7
0868  5E EA 76 7A 6B 82 C0 C5  : AA
0870  01 FF 15 E7 60 62 80 FF  : 3D
0878  84 EC F7 98 A0 F7 85 C7  : E2
-----------------------------------
SUM:  40 B4 C1 F8 CD 0C 09 69  D9F9

0880  3A 92 FA 35 B1 73 14 1D  : 50
0888  E1 78 18 A0 F0 58 4E E6  : 8D
0890  28 3B D2 CF DE 62 81 3F  : 04
0898  64 D9 E4 FE 3B 78 BA 82  : 0E
08A0  C4 3B 24 21 D9 2A C3 B2  : BC
08A8  D6 28 12 8C 50 1B EF C3  : B9
08B0  B1 68 C5 02 B8 B5 D1 52  : 70
08B8  AE 30 16 F0 16 80 2D 40  : E7
08C0  5B 78 05 BA F8 5C 00 5C  : 42
08C8  40 5C AA 16 4E 01 64 02  : 11
08D0  CC 05 BC 05 A0 0B EC D7  : 00
08D8  BE FA 70 73 0B D7 0E C5  : 50
08E0  B5 8A 05 A3 14 06 96 43  : DA
08E8  B2 8C 50 29 D5 D8 D5 33  : 6C
08F0  01 6F 01 68 02 D4 05 B7  : 6B
08F8  1D 1E BD 41 70 01 71 01  : 1C
-----------------------------------
SUM:  4A 8F C7 FE FD 11 8C F3  CA2E

0900  72 A8 5D CC 50 7C 02 C8  : D9
0908  05 98 0B 78 0B 40 17 95  : 17
0910  D8 DE A0 BA C3 B2 D6 28  : 83
0918  16 E6 28 16 C8 76 51 8A  : 53
0920  05 0C 50 28 62 81 43 14  : C3
0928  0A 18 A0 50 C5 02 86 28  : 87
0930  14 31 40 A1 8A 05 0C 50  : 11
0938  28 62 81 43 14 0B F9 31  : 97
0940  41 E9 75 05 7D 4C 50 7A  : 37
0948  DF 09 55 4E D5 BD F1 85  : 93
0950  AE 95 D0 F7 42 DB 50 5C  : D3
0958  C2 AD 87 65 CC 50 35 90  : 3C
0960  EC A3 14 0A C3 3D 18 43  : 08
0968  B1 84 3B 18 43 B1 84 3B  : 3B
0970  18 43 B1 84 3B 18 43 B1  : D7
0978  84 3B 20 21 D9 01 AF 7F  : 08
-----------------------------------
SUM:  79 94 22 E6 25 B2 62 65  1483

0980  C5 8A 0F CD 52 F5 E2 62  : B6
0988  83 DC B0 9A F1 31 41 EE  : FA
0990  57 1D 7D 0C 50 1F 68 5B  : 2F
0998  6A 0B 98 55 B0 EC B9 8A  : 41
09A0  08 59 0E CA 31 40 B1 19  : 74
09A8  E9 11 0E C8 88 76 44 43  : 55
09B0  B2 22 1D 91 10 EC 88 87  : 8D
09B8  64 44 3B 22 21 D9 21 0E  : 2E
09C0  C9 0D 7B C8 53 B3 21 4E  : 8E
09C8  CC 85 3B 32 14 EC CA B4  : 3C
09D0  EC DA C5 02 DC C5 04 6F  : A1
09D8  C3 B1 A8 C5 03 3A B9 E8  : BF
09E0  CE 30 16 F0 16 80 2D 40  : 07
09E8  5B 00 B8 00 B8 80 BA C3  : C8
09F0  B1 89 C0 2C 6B DC B9 A9  : CF
09F8  D9 F4 C3 B1 8B A2 9D 9E  : A9
-----------------------------------
SUM:  07 28 BC 9B 37 C8 C7 C9  75D3

0A00  40 B4 D7 51 2B 71 31 41  : 2A
0A08  EE A7 65 B8 98 A0 F6 D3  : B3
0A10  B2 D6 B1 40 D7 31 41 2B  : ED
0A18  21 D9 46 28 18 C6 7A 18  : D8
0A20  87 61 88 76 18 87 61 88  : 6E
0A28  76 18 87 61 88 76 18 87  : 13
0A30  61 88 76 3C 43 B1 E3 5E  : D0
0A38  EF 1A F7 78 D7 BB C6 BD  : 8D
0A40  DE 35 EF 56 28 1A D6 28  : 98
0A48  1A 8C 50 3D D6 43 B2 8C  : 8A
0A50  50 30 62 81 83 14 0C 18  : 1E
0A58  A0 60 C5 03 06 28 18 31  : 3F
0A60  40 C1 8A 06 0C 50 30 62  : 7F
0A68  81 83 14 0C 18 A0 65 1A  : 5B
0A70  F7 51 AF 75 1A F7 51 AF  : 7D
0A78  75 AC 3B 2D 62 81 A8 C5  : D9
-----------------------------------
SUM:  63 B7 9D C7 93 72 3E 6E  C05A

0A80  03 CA C8 76 51 8A 06 0C  : F8
0A88  50 30 62 81 BF FB 14 0B  : 3C
0A90  88 2F 2E 7A 2E 2F ED 73  : 1C
0A98  78 BE 39 70 AE 60 7D F1  : 5B
0AA0  79 85 E4 CF 45 E1 E5 DE  : 9A
0AA8  8F 1F 98 5D 21 D8 B9 82  : D7
0AB0  FE FF 6B 98 5C 78 FD 8F  : 60
0AB8  2F 8D E3 0B DD 87 62 E8  : 58
0AC0  E7 83 31 F8 98 A0 EB F1  : A7
0AC8  7E 26 28 17 84 4F 8F 31  : 76
0AD0  F7 98 A0 CB E9 79 B3 1E  : 2D
0AD8  E6 28 32 69 0F 2E 61 AE  : F5
0AE0  10 B4 CD EC 6E FC 35 DF  : FB
0AE8  B9 E2 FC B0 EC 5C DF FD  : 6B
0AF0  81 75 87 65 AC 50 35 18  : 2B
0AF8  A0 79 D9 0E CA 31 40 C1  : FC
-----------------------------------
SUM:  B4 04 AF 02 6F 3B 98 F5  9C52

0B00  8A 06 0C 50 30 62 81 83  : 82
0B08  14 0C 18 A0 60 C5 03 06  : 06
0B10  28 18 31 40 C1 8A 06 0C  : 0E
0B18  50 30 62 81 A4 35 EF 21  : 4C
0B20  AF 79 0D 7B C8 6B DE 55  : 16
0B28  87 65 AC 50 35 18 A0 7B  : 50
0B30  EF C3 B2 14 62 82 1E D6  : 50
0B38  28 3E E5 11 26 17 D1 9E  : 08
0B40  87 98 2D FA 42 DD 88 2D  : 1A
0B48  35 B4 93 07 28 0B 6E D6  : FA
0B50  DE 71 85 C3 A5 72 4C A1  : 9B
0B58  71 01 72 D2 17 8B 42 7F  : 19
0B60  8B 14 07 EA E6 61 64 02  : 3D
0B68  CC 05 BC 05 E5 D7 B9 74  : 7B
0B70  85 D2 9D 92 C4 0E 5F 90  : 47
0B78  7C C2 DB 10 5E 4E 7F 0C  : 60
-----------------------------------
SUM:  C6 A4 F9 C8 8D 7B 65 2F  654B

0B80  3B 98 56 08 76 42 8C 50  : C5
0B88  3D 2C 87 65 18 A0 86 F3  : 86
0B90  14 1E C6 57 6E D4 66 F3  : EA
0B98  14 07 FA 07 CC 2F 1E 7A  : AF
0BA0  77 05 A6 70 B3 D0 16 A0  : CB
0BA8  2D B4 05 9E 70 B8 73 85  : A4
0BB0  9E 60 B8 80 B9 67 D0 9E  : C4
0BB8  80 BC 0C 50 1E 8D D4 65  : 7C
0BC0  0B 20 16 60 2D E0 2D 3A  : 15
0BC8  42 CB 8C 2D 79 C2 F7 61  : 59
0BD0  61 6C 48 26 16 D9 C2 FF  : EB
0BD8  18 84 4C 2A D8 76 5C C5  : 81
0BE0  02 D9 0E CA 31 41 04 14  : 3D
0BE8  EC A0 A7 65 05 3B 28 29  : 29
0BF0  D9 41 4E CA 0A 76 50 53  : 55
0BF8  B2 82 9D 94 14 EC A8 A7  : B4
-----------------------------------
SUM:  A1 D5 E2 13 AA 30 29 6E  DCC7

0C00  65 45 3B 2A 29 D9 51 4E  : B0
0C08  CA 8A 76 54 53 B2 B5 A7  : 7F
0C10  66 D6 28 21 73 14 0D 64  : 7D
0C18  3B 28 C5 04 3D 0C 50 76  : 3B
0C20  D2 83 E8 62 83 B6 8F 56  : BD
0C28  78 0B 14 EC B2 D6 98 D6  : 79
0C30  14 EC B0 A7 65 85 3B 2C  : A8
0C38  29 D9 80 A7 66 02 9D 98  : C6
0C40  0A 76 60 29 D9 80 A7 66  : 6F
0C48  02 9D 98 56 9D 9B 58 A0  : BD
0C50  85 CC 50 42 C8 76 6D 31  : BF
0C58  41 E3 F2 78 BE 7F B7 C9  : 4B
0C60  B3 F1 FC 5F C8 00 00 00  : C7
0C68  00 00 00 00 00 00 00 00  : 00
0C70  00 00 00 00 00 00 00 00  : 00
0C78  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  DC D3 00 D7 F0 CE 85 BF  DAD6
```

THE SENTINEL

〈対応機種一覧〉●MZ-80K/C/700/1500●MZ-80B/2000●MZ-2500/2861●X1●X1turbo/Z●PC-8001/8801/88●SMC-777/C●PASOPIA/5●PASOPIA 7●FM-7/77/AV●PC-286/386/9801/98●X68000
掲載されたプログラムの利用には各機種用のS-OS"SWORD"システムが必要です。

第127部　MAKE 実践Small-C講座(8)

●MAKE

以前から載るかもしれないとウワサがあった，MAKEプログラムを発表します。

今月発表するMAKEプログラムは，C MAGAZINEの1991年6月号に掲載されたPMKを移植したもので，機能的にはなかなかシンプルなものです。しかし石上氏が本文中で述べているように，シンプルとはいっても既存のDOSマシンに比べての話ですから，S-OSで使うぶんには問題ありません。

ある手順を一括して行ってくれるバッチファイルも便利ですが，手順に無駄が生じる場合があります。

必要なことを無駄なく行ってくれるMAKEプログラムは，大規模なプログラムを作成するときに大きな力を発揮してくれるでしょう。煩わしい無駄な時間はないほうがいいに決まってますからね。

そして，いままで連載してきた結果がこのMAKEプログラムによって結集したといえるでしょう。

●使いやすいシステムとは？

今月で8回目となる実践Small-C講座も，今回のMAKEをもって一応の区切りをむかえることになりました。

内容的には，主にSmall-Cを中核にして石上氏の思い描くシステム環境を構築してきました。いろいろあるシステムの中で，DOSマシンの操作体系を参考にS-OSを強化してきたものです。

また，これまで行ってきた拡張は，あくまでも一例にすぎません。今回はDOSマシンを参考にしてきましたが，ユーザー独自のまったく新しいシステムでもかまわないでしょう。

マシン語モニタとしてのS-OS"SWORD"ですから，不満を感じることが少なくないかもしれません。しかし，そのぶんパワーさえあれば，自分の作ったプログラムで思いどおりの環境を作れるのです。また，いままで発表されたツールをうまく組み合わせればそれなりのシステムが出来上がるはずです。

実際，投稿されてくるプログラムディスクには，ユーザーによってカスタマイズされたシステムが多く見られます。苦労をしたとしても，思いどおりのシステムができれば，それによって得られる利益ははかり知れないものがあるのです。

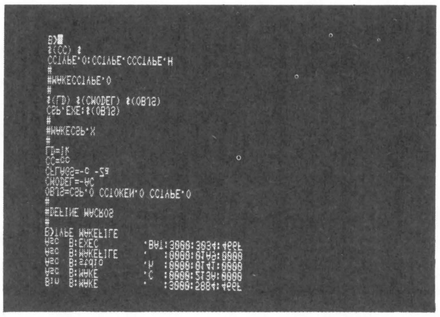
皆さんもがんばってください。

●S-OSの系譜(39)

1990年1月号では，SLANGコンパイラ再掲載，そのSLANG用ゲーム「WORM KUN」が発表されました。

その多機能さもさることながらS-OSのシステムに合った，かゆいところまで手が届く仕様が，ユーザーにとても使いやすいコンパイラと評価されたのでしょう。異例の再掲載となりました。

実際に，SLANGコンパイラの有用性は，



THE SENTINELの歴史を見ることでも証明されています。ゲームからS-OSシステムの拡張，さらにSLANGコンパイラ自身もSLANGコンパイラによって強化され続けているのです。現在でも投稿されてくるものの中で，SLANGコンパイラを使ったものが少なくありません。

もうひとつのSLANGコンパイラ用「WORM KUN」は，"@"である自分が逃げ回るエサ"Q"を食べる，すると自分の体が伸びていき，おじゃまキャラに体が触れてしまうとミス，という古典的なアクションゲーム。エサの捕まえ方がちょっと独特で，いったんスペースキーでポインタを置き，もう一度スペースキーを押すことで，頭とポインタを対角とする四角形の網を張ってエサを捕まえるという，リブルラブルにあったバシシを思い出させるものでした。

それでは，皆さんもまだまだ現役の開発言語といえるSLANGコンパイラを使ってみませんか。ディスクやテープに埋もれさせておくのはもったいないですからね。もちろん，いままで発表された言語でもかまいませんよ。THE SENTINELでは皆さんの投稿をいつでも歓迎します。

1992■インデックス

■92年1月号
第115部　LINER
■92年2月号
第116部　シミュレーションゲームPOLANYI
■92年3月号
第117部　カードゲームKLONDIKE
■92年4月号
第118部　オプティマイザO80実践Small-C講座(1)
■92年5月号
第119部　COMMAND.OBJ実践Small-C講座(2)
■92年6月号
第120部　COMMAND.OBJ2実践Small-C講座(3)
■92年7月号
第121部　関数リファレンス実践Small-C講座(4)
■92年8月号
第122部　ワイルドカード実践Small-C講座(5)
第123部　グラフィックライブラリ GRAPH.LIB
■92年9月号
第124部　O-EDIT＆MODCNV
■92年10月号
第125部　SLENDER HUL実践Small-C講座(6)
■92年11月号
第126部　EDIT実践Small-C講座(7)

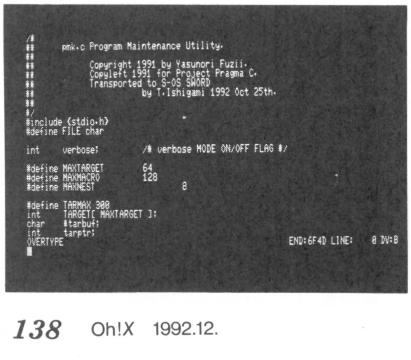
全 機 種 共 通
S-OS"SWORD"要

MAKE

実践Small-C講座(8)

Ishigami Tatsuya
石上 達也

今月は，予告どおりEDITで出力されたファイル情報をもとに動くMAKEプログラムの登場です。これがあれば，面倒な分割コンパイル作業もMAKEが自動的に行ってくれます。



COMMAND.OBJ以来このシリーズでは，“SWORD”の操作性向上を目指す，ということで話を進めてきました。あるいは，MS-DOSやHuman68kなどと同じような操作性を目指してきた，といってもよいでしょう。

いままで7回にわたって連載してきたこの講座も，今回のプログラムをもって最終回を迎えます。別に，目的の環境を実現でき，これ以上やりようがなくなった，というわけではなく，ひととおりやりたいことをやり終えただけの話です。また何か，面白そうなことを思いついたり，リクエストが殺到したりするようなことがあったり，ドラクエVを年内に解き終わってしまったりすれば，このコーナーに戻ってくるかもしれません。

MAKEとは

バッチ処理とは，ある一定の操作手順をひとまとめにして，繰り返し行われる作業を自動的に実現しようとするものでした。

いままで，コンパイル→アセンブル→リンクと1つひとつ手動で行っていた操作が，これらの手順を書き表した手順書（バッチファイル）をコンピュータに指定してやることで，自動的に行うことができるようになりました。

ここで，図1を見てください。

これは，Small-Cを構成するファイルの相関図です。これらのファイルが図1のように複雑に関係し合って，Small-Cは構成されています。

たとえば，cc12.Cというファイルを変更したら，cc1.Cをコンパイルし直さなければいけないということですし，CC.defを変更したときは，cc1.C,cc2.C,cc3.Cを変更し直さなければいけないということになります。

複数のファイルを変更し，そのひとつのファイルの変更に1時間近くもかかっていた場合を考えてみましょう。

なにかのメモを几帳面にとっておける能力のある人は別として，1時間のうちに，自分がどのファイルをいじったかということを覚えておけるのは，かなりの幸運な場合です。たいていは，「1時間前の自分は他人」の世界です。

このような事態に対する手段として真っ先に思い浮かぶのが，すべてを一から構築し直すようなバッチファイルの作成でしょう。しかし，Small-Cを使ったことのある人ならわかると思いますが，ひとつのファイルをコンパイルする（ということは，その先にアセンブルという作業も待っている）ことは，たいへん時間のかかる作業です。しかも，ファイルを分割するということは，通常，大きすぎてまとめて管理できないということですので，それぞれのファイルのサイズもかなり大きいはずです。大きいファイルはコンパイルやアセンブルにもそれだけ長い時間を要します。この時間が手を加えていないプログラムのコンパイルなどに当てられた場合，無駄な時間が多く発生することになります。

そこで，人間がいま何をすればいいのかをファイルの状況から判断して，無駄なコンパイルなどを行わないように配慮しながら，コーディネートしてくれるプログラムがほしくなってきます。それがこのMAKEです。

PMKについて

以前，私がこのコーナーで「そろそろ“SWORD”で使えるMAKEがほしいよー」と書いたら，それに応えて，さっそくこのコーナーの常連さんのひとりである西村進氏が，独自のMAKEを投稿してくれました（実は1991年の話で，けっこう古い話です）。当初は，このプログラムが掲載される方向に編集部は動いていたのですが，これを使っているうちに，MS-DOSやHuman68k用のMAKEとかなり異なった操作方法を持っていることがわかりました。読者のほとんどがX68000を持っている時代に，まったく独自の操作方法を持つMAKEを

今月のバグ取り

え〜と，もうないと思っていたんですが，やっぱりまだまだありました。しかも，puts()という要の関数に。

このputs()という関数は，引数として指定された文字列を画面に表示するというものでしたが，この関数は文字列の表示を行ったあとに，自動的に改行することになっています。しかし，Small-Cでは，そうはなっていません。で，RDRTL.ASMというライブラリファイルの中のラベルfputs3以下を次のように変更してください。この変更を行わないと，今回のプログラムは正常に動作しません。

```
fputs3 : LD     BC, 0DH
         PUSH   BC
         PUSH   DE
         CALL   putc    ; putc('¥n', unit)
         POP    DE
         POP    BC
         LD     HL,0
         RET            ; return(NULL)
```

掲載して，“SWORD”だけが独自の方向に進むのは，少し問題があるのではないか，という考えが頭をもたげてきました。

そんなとき，目についたのが参考文献のPMKでした。このプログラムは作者の藤井保則氏が提唱する「Project Pragma」の一環として作成されました。このProject Pragmaというのは，C MAGAZINE誌上で藤井保則氏を中心とするメンバーによって（ということになっているが，実質的には氏１人ががんばっているようである），まったく新しいCコンパイラを作ろうという企画でした。

ただし，このPMKはフルセットのMAKEなどに比べると，かなりの機能が省略されています。これをMS-DOSへ持っていって，Turbo-CやMS-CについてくるMAKEの代わりに使う場合には，かなりきつい制約となるかもしれませんが，今回のように“SWORD”に移植する場合には，かえって機能が少ないほうが，解析する量も少ないですし（これは，バグの入り込む可能性も少なくなることにつながる），メモリの占有量も少なくてすむので好都合です。

ちなみに，PMKで省略された機能の多くは「扱うファイルが50や100もあるプログラムで，これらの相関関係を記述する際に，あればひょっとしたら役立つかもしれないもの」ですので，あってもなくても，“SWORD”ではあまり関係がありません。

以下では，藤井氏の作成されたMAKEをPMKと記し，それ以外のもの，特に区別する必要がないときには，ただMAKEと記述します。

MAKEのカラクリ

「ファイルの状況から判断して動作をコーディネートする」プログラムといいましたが，「ファイルの状況」については説明していませんでした。

MS-DOSやUNIXには，ファイルの属性（ファイルの名前，ファイルのサイズなど）の中に，そのファイルが作成された日時，あるいは最後に更新された日時に関する情報（専門用語でタイムスタンプといいます）を持っています。MAKEはこれらの情報をもとにファイルの状況を見て，どのように目的のファイルを作成するのかを判断します。

最終的に必要とされるファイルの作成日時より，それを構成するファイルの作成日時のほうがあとであれば，MAKEは自動的にコンパイルやアセンブルを行います。

さて，このようなMAKEをそのまま“SWORD”上に持ってこれればよいのですが，残念ながら“SWORD”は，このファイルの作成された日時を示す情報を持っていません。これでは，どのファイルが新しく更新され，どのファイルには手を加えられていないのかを知るすべがありません。

ここで，先月号のEDIT.OBJの発表の際に説明したプラスαの部分が役立ってきます。EDIT.OBJはファイルのセーブを行う際に，そのファイル名をUPDATE.$$$というファイルに記録しました。ファイルをセーブするということは，そのファイルは更新されたということですから，MAKEはこの情報を使って，ファイルの新旧を判断すればよいわけです。

てなわけで，今回発表するMAKEを使おうと思ったら，

1)　エディタに先月号のEDIT.OBJを用いる
2)　各自のエディタをファイルUPDATE.$$$をアクセスするように改造する
3)　ファイルを更新するたびに，手動でファイルUPDATE.$$$にそのファイル名を登録する

のどれかを行わなければなりません。

定義ファイル

ファイルの新旧関係を知るカラクリはわかりました。次のカラクリ，ファイルの主従関係（どのファイルを変更すると，どのファイルに影響があって……というような

図1　Small-Cのファイル相関図

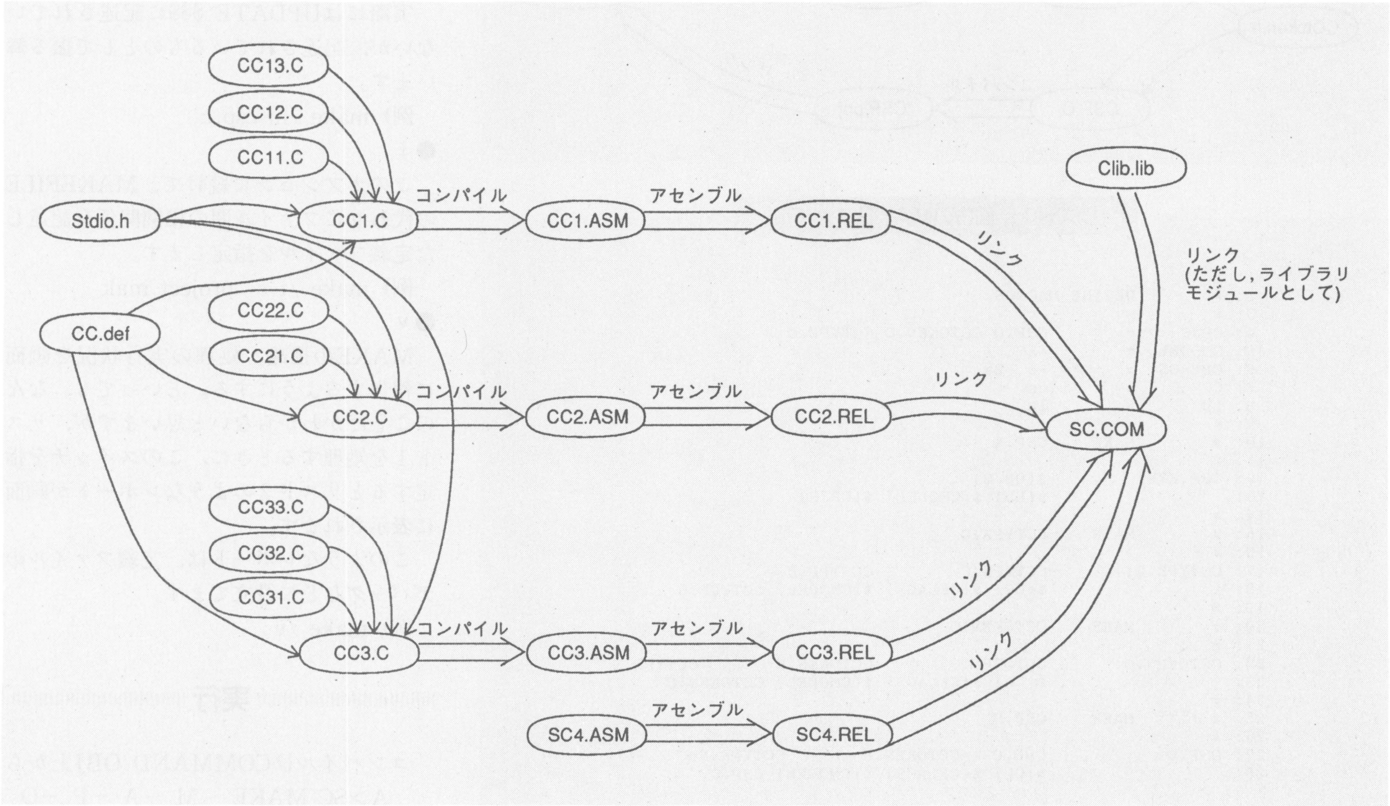


図１の矢印の向き）を見ていきましょう。

さすがに，MAKEが勝手に「拡張子が“.h”はヘッダファイルだから」とか，プリプロセッサよろしくファイルの中身を調べて「おっ，ここに#includeがあって，このファイルを読み込んでいるじゃねぇか」などと判断しているわけではありません。

最近はやりの統合開発環境の中には，そのような優れたものがあるらしいのですが，今回のMAKEはそこまでサポートしていません。人間が各ファイルの主従関係を教え込んでやらなければならないのです。

MAKEに各ファイルの主従関係を教え込んでやるには，カレントデバイス上の“MAKEFILE”という名前のファイルに対し，その情報を書き込んでやります。書き込んでやる際のフォーマットは，図２に示すとおりです。

たとえば，図３に示すようなファイルの相関関係はリスト１のように書き下します。このリスト中で，“#”で始まる行はコメント行と見なされ，MAKEには無視されます。この“#”は，行の先頭から始まっていなければなりません。

また，この定義ファイルの名前を変えたい場合には，MAKEの起動時にコマンドラインからパラメータとして，変えたい名前を指定します。

例）デバイスC：にあるPROJECT.MAKを定義ファイルとするとき。

A>MAKE /f C：PROJECT.MAK

図2　MAKEFILEのフォーマット

```
ターゲットファイル名：　依存ファイル名,
 [依存ファイル名]……
            ターゲットを作成するためのコマ
            ンドライン
# コメント
```

マクロ

現在のMAKEには非常に簡単なマクロ機能が用意されています。マクロ機能というのは，Small-Cでも用意されていた#define命令のように，特定の文字列を別の文字列に置き換えて解釈する機能のことです。

マクロ定義は以下の書式で宣言されます。

［書式］

　macroname ＝ macrostring

［文例］

　cc＝cl －c

たとえば，リスト１では以下のような置き換えが行われます。

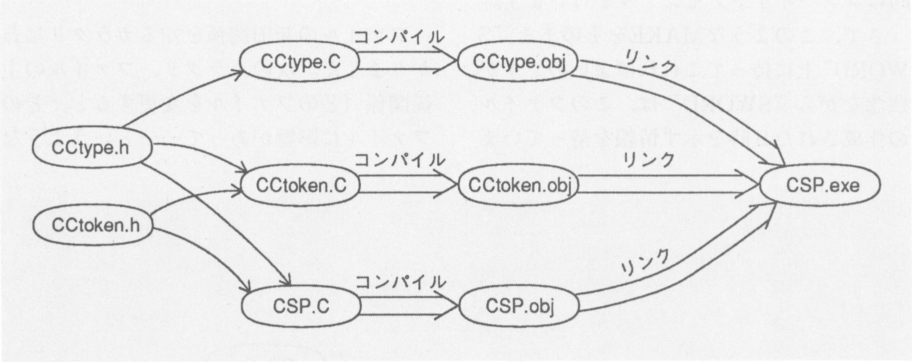
［記述］

```
csp.exe：$(OBJS)
         $(CC) $(CMODEL)
         $(OBJS)
```

［展開］

```
csp.exe：csp.obj cctoken.obj cctype.
         obj
         cl -AC csp.obj cctoken.ob
         j cctype.obj
```

図3　ファイル相関図の例



リスト1　図3のMAKEファイル

```
#
#       DEFINE MACROS
#
OBJS    =       CSP.O CCTOKEN.O CCTYPE.O
CMODEL  =       -AC
CFLAGS  =       -c -Za
CC      =       cc
LD      =       lk
#
#       MAKE    CSP.X
#
CSP.EXE:        $(OBJS)
                $(LD) $(CMODEL) $(OBJS)
#
#       MAKE    CCTYPE.O
#
CCTYPE.O:       CCTYPE.C        CCTYPE.H
                $(CC) $(CFLAGS) $(CMODEL) CCTYPE.C
#
#       MAKE    CCTOKEN.O
#
CCTOKEN.O:      CCTOKEN.C       CCTOKEN.H       CCTYPE.H
                $(CC) $(CFLAGS) $(CMODEL) CCTOKEN.C
#
#       MAKE    CSP.O
#
CSP.O:          CSP.C   CCTOKEN.H       CCTYPE.H
                $(CC) $(CFLAGS) $(CMODEL) CSP.C
```

オプション

MAKEでは，起動時にいくつかのオプションを指定することができます。オプションは，コマンドライン上で“/”または“－”のあとに，オプション名を表す１文字で指定します。オプションは大文字でも小文字でもかまいません。

●h

MAKEで使えるオプションの一覧表を表示します。

例）make /h

●m

実際にはUPDATE.$$$に記述されていないが，記述されているものとして振る舞います。

例）make /m csp.c

●f

このオプションに続けて，MAKEFILEの代わりにファイル間の相関関係を記述した定義ファイルを指定します。

例）make /f c：project.mak

●v

MAKEの推論，処理の実行状況を画面に報告するようにする。といっても，なんのことだかわからないと思いますが，リスト１を処理するときに，このスイッチを指定するとリスト２のようなレポートが画面に表示されます。

このようなレポートは，定義ファイルのデバッグなどに役立てます。

例）make /v

実行

コンパイルはCOMMAND.OBJ上から，

A>SC MAKE －M －A －P －O

A>WZD =MAKE

A> WLK /P:3000,MAKE,clib/S,MAKE/P

とすればOKです。

さて，MAKEを走らせます。コンピュータがファイルの相関関係やファイルの新旧を調べます。そして，現在，どのファイルがコンパイルやアセンブルをし直せばよいのかを判断します。そして，それを実行させます。

と，MS-DOSなどではうまくいくのですが，残念ながら“SWORD”ではそうなりません。MS-DOSでは，あるプログラムから，ファイル上のもうひとつのプログラムをサブルーチンのように呼び出して（たとえ，コンパイラやエディタでも)，また元のプログラムに戻ってくる「チャイルドプロセス」という機能がサポートされていて，MAKEもコンパイラやらアセンブラやらを呼び出すのですが，“SWORD”にはそういう機能がありません。

COMMAND.OBJのときのように，まだどこかから空きエリアを見つけてきてそこに埋め込むという手もあるのですが，狭い64Kバイトのメモリ空間で，これ以上このような手は使いたくありません。

そこで，どうするかというと，MAKEのプログラム内でそれらのプログラムの実行をあきらめてしまうのです。つまり，MAKEは，このあとに行うべき動作のコーディネートのみに撤し，実行部はほかのプログラムに任せてしまうのです。

具体的には，MAKEは出力として，EXEC.BATという名前のバッチファイルを作成します。このバッチファイルの中に，何を行えば目的のプログラムが作成されるのかというようなことが書いてあります。つまり，MAKEを実行したあとで，このバッチファイルを動かしてやらなければなりません。今回のMAKEでは正常終了時に，画面上に“EXEC.BAT”と表示しますので，リターンキーを押すだけで，このバッチファイルを実行することができます。

手抜き

例によって，手抜きというか，仕様が完璧に満たない部分が多少残っています。

●1枚のフロッピーディスクにひとつのターゲットしか設定できない

これは，ファイルの更新情報を管理するファイルUPDATE.$$$の扱い方の問題です。先ほども述べましたが，今回のMAKEはファイルの新旧情報を得る手段としてタイムスタンプ機能を利用せずに，このファイルのみに頼っています。で，1回MAKEを実行させるたびに，このファイルを消去してしまいます。

ということは，ひとつのフロッピーディスクの中に2つ以上の（最終的なオブジェクトファイルとしての）ターゲットを設定した場合，Aというファイルを作成するためにMAKEを実行すると，Bというファイルに関する情報は，すべて消え去ってしまうのです。

最初は，UPDATE.$$$をすべて消去するなどという乱暴な手段は用いずに，MAKEが利用した情報だけそこから抜き取って……などと考えていたのですが，ひとつのファイルを2つのターゲットから利用していた場合はどうなるんだとか，もうひとつのターゲットの主従関係情報の収められたファイルネームは，どこから持ってくるんだ，などの難しい問題もあって，今回はずばっと見送りました。つまり，1枚のフロッピーディスク（ハードディスク，RAMディスクでも，もちろん事情は同じ）には，ひとつのターゲットしか設定できません。

でも，Small-C本体の場合を見てもらえれば，おおよその状況は想像できるのではないかと思うのですが，2D(=360Kバイト）のフロッピーディスクには，たぶんひとつのターゲットプログラムに関係したファイルしか収められないのではないかと思います。

●実行中のエラー中断はサポートされていない

本家本元のUNIXではどうなっているのか知らないのですが，MS-DOSやHumanのMAKE では，コンパイラやアセンブラが実行中にエラーを起こしたら，そこで一度処理を打ち切って，プログラマがソースリストを見直すというようなことができるようになっています。なぜ，わざわざ，そんなことを書いたかというと，そう，今回のMAKEではできないのです。もう少し細かくいうと，これはバッチ処理を担当しているCOMMAND.OBJに，ひいては“SWORD”の規格自身に問題があります。

MS-DOSなどでは，プログラム自身も

リスト2 Vスイッチ使用時のレポート

```
Program MaKe utility version 1.0
Copyleft 1991 by ProjectPragmaC.
Define $(OBJS) to "CSP.O CCTOKEN.O CCTYPE.O"
Define $(CMODEL) to "-AC"
Define $(CFLAGS) to "-c -Za"
Define $(CC) to "cc"
Define $(LD) to "lk"
Target CSP.EXE: Dependent "CSP.EXE" "CSP.O" "CCTOKEN.O" "CCTYPE.O"
                by [lk -AC CSP.O CCTOKEN.O CCTYPE.O]
Target CCTYPE.O: Dependent "CCTYPE.O" "CCTYPE.C" "CCTYPE.H"
                by [cc -c -Za -AC CCTYPE.C]
Target CCTOKEN.O: Dependent "CCTOKEN.O" "CCTOKEN.C" "CCTOKEN.H" "CCTYPE.H"
                by [cc -c -Za -AC CCTOKEN.C]
Target CSP.O: Dependent "CSP.O" "CSP.C" "CCTOKEN.H" "CCTYPE.H"
                by [cc -c -Za -AC CSP.C]
TARGET#01 "CSP.EXE"
dependent "CSP.O"... is TARGET#04
          TARGET#04 "CSP.O"
          dependent "CSP.C"
          dependent "CCTOKEN.H"
          dependent "CCTYPE.H"
          MAKING#04 "CSP.O"
cc -c -Za -AC CSP.C
dependent "CCTOKEN.O"... is TARGET#03
          TARGET#03 "CCTOKEN.O"
          dependent "CCTOKEN.C"
          dependent "CCTOKEN.H"
          dependent "CCTYPE.H"
          MAKING#03 "CCTOKEN.O"
cc -c -Za -AC CCTOKEN.C
dependent "CCTYPE.O"... is TARGET#02
          TARGET#02 "CCTYPE.O"
          dependent "CCTYPE.C"
          dependent "CCTYPE.H"
          MAKING#02 "CCTYPE.O"
MAKING#01 "CSP.EXE"
lk -AC CSP.O CCTOKEN.O CCTYPE.O
TARGET#02 "CCTYPE.O"
dependent "CCTYPE.C"
dependent "CCTYPE.H"
MAKING#02 "CCTYPE.O"
TARGET#03 "CCTOKEN.O"
dependent "CCTOKEN.C"
dependent "CCTOKEN.H"
dependent "CCTYPE.H"
MAKING#03 "CCTOKEN.O"
TARGET#04 "CSP.O"
dependent "CSP.C"
dependent "CCTOKEN.H"
dependent "CCTYPE.H"
MAKING#04 "CSP.O"
```

ひとつの戻り値を持っていて，この値が負の数であったら，なんらかの不都合が生じたことを表すことになっています。しかし，“SWORD”のホットスタート機能（この機能を使って，プログラムの終了とする）のところには，何も引数が指定されていません。COMMAND.OBJにて“SWORD”を改造するときに，勝手にこの引数を決定してもよかったのですが，いままでのアプリケーションを使用する場合に不都合を生じそうなのでやめておきました。てなわけで，MAKEの実行中は（細かいことをいえば，MAKEの作成したバッチファイルの実行中は）なるべくコンピュータの側を離れないで，一連の作業中に何が起こったかを確認していてください。

●定義ファイル中で“#”を途中から書いてはいけない

これは，おおもとのPMKが対応していなかったのですが，対応しても利用価値が少ないので手を加えてありません。

＊　＊　＊

実は，このプログラムのデバッグはX68000で行いました。例によってC言語の持つ「移植性」の高さに思いっきり支えられたのでした。

最後にこのプログラムを移植するにあたって，快く承諾をしてくださったC MAGAZINE編集部と，PMK作者の藤井氏に感謝します。

<参考文献>

藤井保則，PMKの作成，C MAGAZINE1991年6月号

リスト3

```
/*
**      pmk.c   Program Maintenance Utility.
**
**              Copyright 1991 by Yasunori Fuzii.
**              Copyleft 1991 for Project Pragma C.
**              Transported to S-OS SWORD
**                      by T.Ishigami 1992 Oct 25th.
**
*/
#include <stdio.h>
#define FILE char

int     verbose;            /* verbose MODE ON/OFF FLAG */

#define MAXTARGET       64
#define MAXMACRO        128
#define MAXNEST         8

#define TARMAX 300
int     TARGET[ MAXTARGET ];
char    *tarbuf;
int     tarptr;

#define COMMAX  500
int     COMMAND[MAXTARGET];
char    *combuf;
int     comptr;

/*  char    *updateFilename[MAXTARGET]; */
int     updateFilename[MAXTARGET];

int     TARGETCOUNT;

/*  char    *mcname[MAXMACRO]; */
/*  char    *mcline[MAXMACRO]; */
int     mcname[MAXMACRO];
int     mcline[MAXMACRO];

int     macrocount;
int     nestlevel;

char    makefilename[80];       /* "MAKEFILE" */
FILE    *fpo;   /* Filedescription of "EXEC.BAT" */

main(argc, argv) int argc; char *argv[]; {
    int   n;

    verbose =
    tarptr =
    comptr =
    TARGETCOUNT =
    macrocount =
    nestlevel = 0;

    tarbuf = malloc(TARMAX);
    combuf = malloc(COMMAX);

    puts("Program Make utility version 1.0");
    puts("Copyleft 1991 by ProjectPragmaC.");

    strcpy( makefilename, "MAKEFILE" );
    makeflags( argc, argv );
    makeChgFileList();
    loadmakefile( makefilename );

    nestlevel = 0;
    fpo = fopen("EXEC.BAT" , "w" );
    if( fpo == NULL ) {
        puts("Can't Open EXEC.BAT");
        abort();
    }
    for( n = 0; n < TARGETCOUNT; n++ )
        maketargetfile( n );
    fclose(fpo);
    delete("UPDATE.$$$");
    puts("  EXEC.BAT¥36");
}

/*
**      Make Option Flags.
*/
makeflags(argc, argv)  int argc; char *argv[]; {
    int     argn;
    char    *cp;

    for( argn = 1; argn < argc; argn++ ) {
        cp = argv[argn];
        if(*cp == '-' || *cp == '/') {
            if(toupper(*(++cp)) == 'V')  {
                verbose = 1;
            } else if(toupper(*cp) == 'F') {
                strcpy( makefilename, argv[++argn] );
            } else if(toupper(*cp) == 'M') {
                recordfile(argv[++argn]);
                argn++;
            } else {
                usage();
            }
        }
    }
}

/*
**      Print Usage
*/
usage( ) {
    puts( "Usage: MAKE [-options]");
    puts( "    v  verbose mode on");
    puts( "    f  filename indicate insted of MAKEFILE");
    puts( "    m  filename regard file as updated");
    puts( "    h  help");
    exit( 1 );
}

/*
**      print error messages and exit.
*/
error(message)  char *message; {
    printf("¥nERROR: %s¥n", message );
    exit( -1 );
}

/*
**
**      load MAKEFILE
**
*/
loadmakefile(fname) char *fname; {
FILE *fp;
char cc, *cp, *pre, *next , buffer[ 256 ], line[ 256 ];
int targetnumber, n;
char *linkpointer;

    targetnumber = 0;
    linkpointer = NULL;

    fp = fopen(fname, "r" );
    if( fp == NULL ) {
        printf("Can't Open =%s¥n", fname);
        abort();
    }

    while(!feof(fp)) {
        if(( cp = fgets( buffer, 256, fp)) == NULL) break;
        while(*cp) {
            if( *cp == '¥n' ) { *cp = 0; break; }
            cp++;
        }
        if( buffer[0] == '#' ) continue;

        while( expand_macro( line, buffer ) );

        if( line[ 0 ] > ' ' )
            set_tm( line );
        else
            set_command( line );
    }
    fclose( fp );
}

/*
** Set target or macro
*/
set_tm(line) char *line; {
char name[ 256 ];
char *cp, *dp;

    cp = line;
    dp = name;

    while( *cp > ' ' ) {
        if( *cp == ':' ) break;
        if( *cp == '=' ) break;
        *dp++ = *cp++;
    }
    *dp = 0;

    while( *cp <= ' ' && *cp) cp++;

    switch( *cp++ ) {
        case ':':
         if( TARGETCOUNT >= MAXTARGET ) error( "TOO MANY TARGET" );
                set_target(name, cp);
                break;
        case '=':
                if( macrocount >= MAXMACRO ) error( "TOO MANY MACRO" );
                set_macro( name, cp );
                break;
        default:
                error( "SYNTAX ERROR... Expecting ':' or '='" );
    }
}

/*
**      define MACORS
*/
```

```
set_macro(name, line)  char *name,*line; {
char *cp;
int n;

    while( *line <= ' ' )
        if( ! *line++  ) error( "SYNTAX... Missing MACROSTRING" );

    n = strlen( line );
    if( ( cp = malloc( n + 1 ) ) == NULL ) error( "MEMORY ERROR" );
    strcpy( cp, line );
    mcline[macrocount] = cp;

    n = strlen( name );
    if( ( cp = malloc( n + 1 ) ) == NULL ) error( "MEMORY ERROR" );
    strcpy( cp, name );
    mcname[macrocount] = cp;

    if( verbose )
        printf( "Define $(%s) to ¥"%s¥"¥n",
                mcname[macrocount], mcline[macrocount]);

    macrocount++;
}

/*
**      Target Listの登録
*/
set_target(name, line) char *name, *line; {
int len;
char *cp, *sp, temp[128];
char *ptr;

    TARGET[TARGETCOUNT] = tarptr;
    len = strlen(name) + 1;
    if(len + tarptr >= TARMAX) error("MEMORY / COMMAND");
    strcpy(&tarbuf[tarptr], name );
    tarptr += len;
    tarbuf[tarptr] = EOF;

    cp = line;
    while( *cp ) {
        while( *cp <= ' ' && *cp) cp++;
        if(!*cp) break;
        sp = temp;
        while( *cp > ' ' )
            *sp++ = *cp++;
        *sp = 0;

        len = strlen(temp)+1;
        if(len + tarptr >= TARMAX) error("MEMORY / COMMAND");
        strcpy(&tarbuf[tarptr], temp);
        tarptr += len;
        tarbuf[tarptr] = EOF;
    }
    tarptr++;

    if( verbose ) {
        ptr = &tarbuf[TARGET[TARGETCOUNT]];
        printf("Target %s: Dependent", ptr);
        while(*ptr != EOF) {
            printf( " ¥"%s¥"", ptr);
            while(*ptr++);
        }
        putchar('¥n');
    }

    combuf[comptr++] = EOF;
    COMMAND[TARGETCOUNT] = comptr;
    TARGETCOUNT++;
}

/*
**      command lineの登録
*/
set_command(line) char *line; {
char *cp;
int len;

    while(*line <= ' ')
        if(!*line++) return;

    if( verbose )
        printf( "                   by [%s]¥n", line );

    len = strlen(line)+1;
    if(len + comptr >= COMMAX) error("MEMORY / COMMAND");
    strcpy(&combuf[comptr], line);
    comptr += len;
    combuf[comptr] = EOF;
}

/*
**      expand macro
*/
expand_macro(cp, sp) char *cp, *sp; {
char cc, *mp, macroname[ 256 ];

    while(cc = *sp++ ) {

        if( cc == '$' && *sp  == '(') {
                sp++;
                mp = macroname;
                while( ( cc = *sp++ ) != ')' ) {
                        if( ! cc ) error( "BAD MACRO" );
                        *mp++ = cc;
                }
                *mp = 0;
                cp = convert_macro( cp, macroname );
                if( cp == NULL ) error( "Undefined MACRO" );
        } else {
                *cp++ = cc;
        }
    }
    *cp = 0;
    return 0;
}

convert_macro(cp, name) char *cp, *name; {
int n;
char *sp;

    for( n = 0; n < macrocount; n++ ) {
        if(!strcmp(mcname[n], name)) {
            sp = mcline[n];
            while( *cp = *sp++ ) cp++;
            return cp;
        }
    }
    return NULL;
}


maketargetfile(number) int number; {
char *ptr, *makename;
char vertab[MAXNEST+1];
int n;

    if( verbose ) {
        if( nestlevel ) memset( vertab, '¥t', nestlevel );
        vertab[nestlevel] = 0;
    }

    ptr = &tarbuf[TARGET[number]];
    if( verbose )
        printf( "%sTARGET#%02d ¥"%s¥"¥n", vertab, number + 1, ptr);

    while(*ptr != EOF) {
        while(*ptr++) ;
        if(*ptr == EOF) break;
        if( verbose )
            printf( "%sdependent ¥"%s¥"", vertab, ptr);
        if(( n = istargetfile(ptr)) < 0 ) {
            if( verbose )
                putchar('¥n');
        } else {
            if( verbose )
                printf( "... is TARGET#%02d¥n", n + 1 );
            if( nestlevel >= MAXNEST ) error( "TOO MANY NESTING" );
            nestlevel++;
            maketargetfile( n );
            nestlevel--;
        }
    }

    ptr = &tarbuf[TARGET[number]];
    makename = ptr;
    if( verbose )
        printf( "%sMAKING#%02d ¥"%s¥"¥n", vertab, number + 1, makename

    while(*ptr != EOF) {
        while(*ptr++) ;
        if(*ptr == EOF) break;
        if(serchfile(ptr) != -1) {
                update(number);
                recordfile(makename);
        }
    }
}

memset(s, c, cnt) char *s; char c; int cnt; {
   while(cnt) {
      s[--cnt] = c;
   }
}

istargetfile(name) char *name; {
int n;

    for( n = 0; n < TARGETCOUNT; n++ ) {
        if(!strcmp(name, &tarbuf[TARGET[n]])) return n;
    }
    return -1;
}


serchfile(name) char *name; {
int      n;
    for( n = 0; n < MAXTARGET; n++ ) {
        if(updateFilename[n] == NULL)              return(-1);
        if(!strcmp(name, updateFilename[n]))     return(n);
    }
}

makeChgFileList() {
    FILE *fp;
    char *cp, buffer[ 256 ];

    fp = fopen("UPDATE.$$$", "r" );
    if( fp == NULL ) {
        puts("Can't Open : UPDATE.$$$");
        abort();
    }

    while(!feof(fp)) {
        if(( cp = fgets( buffer, 256, fp)) == NULL) break;
        while(*cp) {
            if( *cp == '¥n' ) { *cp = 0; break; }
            cp++;
        }
        recordfile(buffer);
    }
    fclose( fp );
}


recordfile(name ) char *name; {
int      n;
    for( n = 0; n < MAXTARGET; n++ ) {
        if(updateFilename[n] == NULL) {
            updateFilename[n] = malloc(strlen(name) + 1);
            strcpy(updateFilename[n], name);
            return;
        }
        if(!strcmp(updateFilename[n], name)) {
            return;      /* It is already exist */
        }
    }
    error("TOO MANY TARGETS");
}

/*
**
**      Execute commandline and update file.
**
**
*/
update(number) int number; {
char *ptr;

    ptr = &combuf[COMMAND[number]];
    while(*ptr != EOF)  {
        if(*ptr != 1) fputs(ptr, fpo);
        while(*ptr) *ptr++ = 1;
        if(*ptr == 0) ptr++;
    }
}
```

▶ 全 機 種 共 通 システムインデックス ◀

＊以下のアプリケーションは，基本システムであるS-OS "MACE" またはS-OS "SWORD" がないと動作しませんのでご注意ください。

1985

■85年6月号
序論　共通化の試み
第1部　S-OS "MACE"
第2部　Lisp-85インタプリタ
第3部　チェックサムプログラム
■85年7月号
第4部　マシン語プログラム開発入門
第5部　エディタアセンブラZEDA
第6部　デバッグツールZAID
■85年8月号
第7部　ゲーム開発パッケージBEMS
第8部　ソースジェネレータZING
■85年9月号
インタラプト　S-OS番外地
第9部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表

1986

■86年1月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年2月号
第15部　S-OS "SWORD"
第16部　Prolog-85入門(1)
■86年3月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年4月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年5月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年6月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ＆エディタ
第24部　"SWORD" 2000 QD
連載　対話で学ぶmagiFORTH
特別付録　PC-8801版S-OS "SWORD"
■86年7月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版S-OS "SWORD"
■86年8月号
第26部　対局五目並べ
第27部　MZ-2500版S-OS "SWORD"
■86年9月号
第28部　FuzzyBASIC発表
連載　明日に向かってmagiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタDREAM
第31部　FuzzyBASIC料理法<1>
■86年11月号
第32部　パズルゲームHOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC料理法<2>
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC料理法<3>

1987

■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC料理法<4>
■87年2月号
第36部　アドベンチャーゲームMARMALADE
第37部　テキアベ作成ツールCONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツールMAGE
付録　"SWORD" 再掲載とMAGICの標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS "SWORD" 変身セット
第43部　MZ-700用 "SWORD" をQD対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASICコンパイラ
第45部　エディタアセンブラZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージJACKWRITE
特別付録　FM-7/77版S-OS "SWORD"
■87年9月号
第49部　リロケータブル逆アセンブラInside-R
特別付録　PC-8001/8801版S-OS "SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASICコンパイラの拡張
第52部　X1turbo版S-OS "SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OSの仲間たち
第53部　もうひとつのFuzzyBASIC入門
第54部　ファイルアロケータ＆ローダ
インタラプト　S-OSこちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo版 "SWORD" アフターケア
　　　　ラインプリントルーチン
特別付録　PASOPIA7版S-OS "SWORD"

1988

■88年1月号
第58部　FuzzyBASICコンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲームELFES
■88年3月号
第60部　構造型コンパイラ言語SLANG
■88年4月号
第61部　デバッギングツールTRADE
第62部　シミュレーションウォーゲームWALRUS
■88年5月号
第63部　シューティングゲームELFES II
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語SLANG入門(1)
第66部　Lisp-85用NAMPAシミュレーション
■88年7月号
第67部　マルチウィンドウドライバMW-1
連載　構造化言語SLANG入門(2)
■88年8月号
第68部　マルチウィンドウエディタWINER
■88年9月号
第69部　超小型エディタTED-750
第70部　アフターケアWINERの拡張
■88年10月号
第71部　SLANG用ファイル入出力ライブラリ
第72部　シューティングゲームMANKAI
■88年11月号
第73部　シューティングゲームELFES IV
■88年12月号
第74部　ソースジェネレータSOURCERY

1989

■89年1月号
第75部　パズルゲームLAST ONE
第76部　ブロックゲームFLICK
■89年2月号
第77部　高速エディタアセンブラREDA
特別付録　X1版S-OS "SWORD"<再掲載>
■89年3月号
第78部　Z80用浮動小数点演算パッケージSOR
　　　　OBAN
■89年4月号
第79部　SLANG用実数演算ライブラリ
■89年5月号
第80部　ソースジェネレータRING
■89年6月号
第81部　超小型コンパイラTTC
■89年7月号
第82部　TTC用パズルゲームTICBAN
■89年8月号
第83部　CP/M用ファイルコンバータ
■89年9月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲームPUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリDIO.LIB

1990

■90年1月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年2月号
第89部　超小型コンパイラTTC++
■90年3月号
第90部　超多機能アセンブラOHM-Z80
■90年4月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部　インタプリタ言語STACK
■90年6月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲームSQUASH!
第95部　X68000対応S-OS "SWORD"
特別付録　PC-286対応S-OS "SWORD"
■90年7月号
第96部　リロケータブルアセンブラWZD
■90年8月号
第97部　リンカWLK
■90年9月号
第98部　BILLIARDS
■90年10月号
第99部　ライブラリアンWLB
■90年11月号
第100部　タブコード対応エディタEDC-T
■90年12月号
第101部　STACKコンパイラ

1991

■91年1月号
第102部　ブロックアクションゲームCOLUMNS
■91年2月号
第103部　ダイスゲームKISMET
■91年3月号
第104部　アクションゲームMUD BALLIN'
■91年4月号
第105部　SLANG用カードゲームDOBON
■91年5月号
第106部　実数型コンパイラ言語REAL
■91年6月号
第107部　Small-C処理系の移植
■91年7月号
第108部　REALソースリスト編
■91年8月号
第109部　Small-Cライブラリの移植
■91年9月号
第110部　SLANG用NEWファイル出力ライブラリ
■91年10月号
第111部　Small-C活用講座（初級編）
■91年11月号
第112部　Small-C活用講座（応用編）
第113部　MORTAL
■91年12月号
第114部　Small-C SLANGコンパチ関数

バックナンバー案内

ここには1991年12月号から1992年11月号までをご紹介しました。現在1991年1，5，8，9，11，12，1992年1，4～11月号の在庫がございます。バックナンバーおよび定期購読の申し込み方法については，178ページを参照してください。

1991

12月号
特集　音・そして音楽とコンピュータ
別冊付録　X68000 THE GAME SOFTWARE BEST SELECTION
連載　響子 in CGわ～るど/マシン語プログラミング/ショートプロ
ハード工作/Z80's Bar/ようこそC言語/ANOTHER CG WORLD
吾輩はX68000である/Computer Music入門/大人のためのX68000
●エレクトロニクスショウ＆データショウ
LIVE in '91　OH YEAH!/サイレント・イヴ/ジングルベル
THE SOFTOUCH　フェアリーランドストーリー/プロサッカー68 他
全機種共通システム　Small-C用　SLANGコンパチ関数

1992

1月号
特集　SX-WINDOWの未来
連載　響子 in CGわ～るど/DōGA・CGA/大人のためのX68000
ハード工作/Z80's Bar/ショートプロ/吾輩はX68000である
ANOTHER CG WORLD/Computer Music入門/カードゲーム
●MAGIC用ゲーム　3D MAZE
●CM-300/500&LA音源の活用法
LIVE in '92　DRAGON SABER/すき/THE ENTERTAINER
THE SOFTOUCH　出たな!! ツインビー/ブリッツクリーク/飛翔鮫 他
全機種共通システム　パズルゲームLINER

2月号(品切れ)
特集　2Dグラフィックの拡張
連載　響子 in CGわ～るど/大人のためのX68000/マシン語プログラミング
ハード工作/ショートプロ/ANOTHER CG WORLD/Z80's Bar
吾輩はX68000である/Computer Music入門/カードゲーム
●TREND ANALYSIS
●MIRAGE MODEL STUFF/Press Conductor PRO-68K
LIVE in '92　ストリートファイターII/Tide Over
THE SOFTOUCH　ジェノサイド2/アルシャーク/コード・ゼロ 他
全機種共通システム　シミュレーションゲームPOLANYI

3月号（品切れ）
特集　SCSIの活用
連載　響子 in CGわ～るど/DōGA・CGA/大人のためのX68000/Z80's Bar
ショートプロ/吾輩はX68000である/マシン語プログラミング
ハード工作/ANOTHER CG WORLD/Computer Music入門/カードゲーム
●Z-MUSIC支援ツール　ZPDCON.X
●Z's-EX用拡張コマンド　MASK_reverse.X
LIVE in '92　ギャラクシーフォース/君が代
THE SOFTOUCH　グラディウスII/レミングス/大戦略III'90/伊忍道
全機種共通システム　カードゲームKLONDIKE

4月号
特集　成熟するゲームと日本の文化
連載　よい子のSX-WINDOW/Z80's Bar
響子 in CGわ～るど/ショートプロ/吾輩はX68000である
ハード工作/ANOTHER CG WORLD/Computer Music入門
●発表　1991年度GAME OF THE YEAR
●バーコードバトラー
LIVE in '92　あじさいのうた/ショパン練習曲作品25-2へ短調/IT'S MAGIC
THE SOFTOUCH　ファーストクィーンII/マスターオブモンスターズII 他
全機種共通システム　実践Small-C(1)オプティマイザO80

5月号
特集　明日のための環境づくり
第7回　言わせてくれなくちゃだワ
連載　響子 in CGわ～るど/大人のためのX68000/Z80's Bar
ハード工作/ショートプロ/マシン語プログラミング
Computer Music入門/吾輩はX68000である
●製品紹介　MIDI音源 03R/W/MIC68K
LIVE in '92　フレンズ/Danger Line
THE SOFTOUCH　エイリアンシンドローム/苦胃頭捕物帳 他
全機種共通システム　実践Small-C(2)COMMAND.OBJ

6月号
特別企画　Oh!MZ,Oh!X10年間の歩み
特別付録　創刊10周年記念PRO-68K(5″2HD)
連載　響子 in CGわ～るど/大人のためのX68000/マシン語プログラミング
ハード工作/ショートプロ/ANOTHER CG WORLD/Z80's Bar
吾輩はX68000である/Computer Music入門
●新製品紹介　Z'sSTAFF PRO-68K ver.3.0
LIVE in '92　Shake the Street/Ancient relics
THE SOFTOUCH　スピンディジーII/ロイヤルブラッド/ライフ&デス 他
全機種共通システム　実践Small-C講座(3)COMMAND.OBJ2

7月号
特集　超空間美術論
特別付録　DōGA CGAシステム＆お試しディスク(5″2HD)
連載　よいこのSX-WINDOW/響子 in CGわ～るど/Z80's Bar
ANOTHER CG WORLD/大人のためのX68000
Computer Music入門/ハード工作/ショートプロ
●試用レポート　V70アクセラレータボード
LIVE in '92　Bye Bye My Love/MATERIAL GIRL/ヴェクザシオン
THE SOFTOUCH　将棋聖天&棋太平68K/シムアース/太閤立志伝
全機種共通システム　実践Small-C講座(4)関数リファレンス

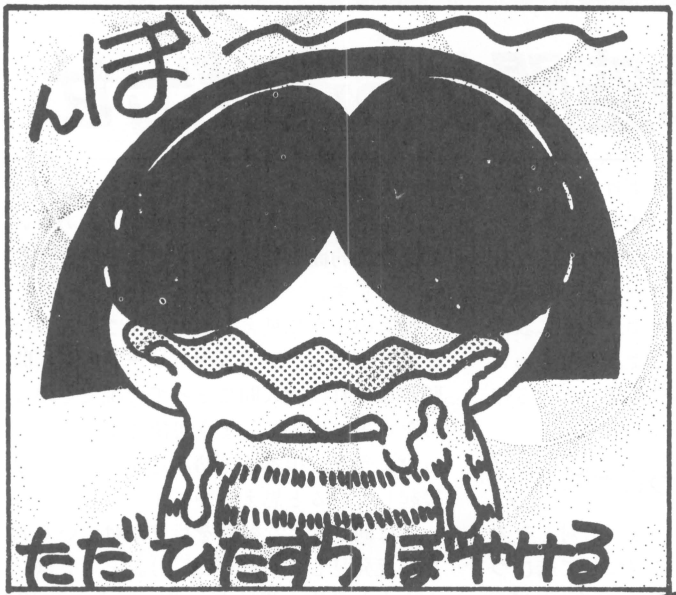
8月号
特集　プログラミング再入門
連載　響子 in CGわ～るど/吾輩はX68000である/よいこのSX-WINDOW
マシン語プログラミング/ハード工作/ANOTHER CG WORLD
大人のためのX68000/Computer Music入門/ショートプロ
●新製品紹介　MATIER/TG100/SOUND SX-68K
LIVE in '92　氷穴/ガラガラヘビがやってくる/風の贈り物
THE SOFTOUCH　三國志III/シムアース/ウルティマVI/バトルテック
全機種共通システム　実践Small-C講座(5)ワイルドカード
グラフィックライブラリGRAPH.LIB

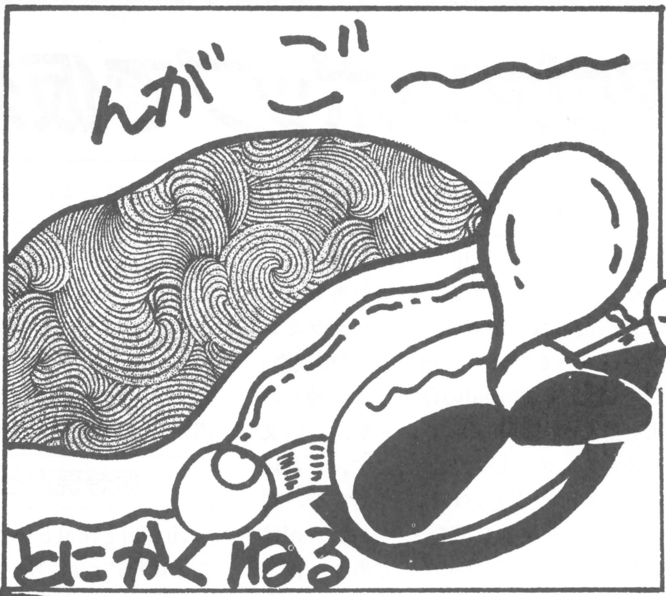
9月号
特集　数値演算の熱い逆襲
連載　DōGA CGアニメーション講座/大人のためのX68000
響子 in CGわ～るど/吾輩はX68000である/ショートプロ
マシン語プログラミング/ハード工作/ANOTHER CG WORLD
●新製品紹介　MATIER/MIREGE System Model Stuff
LIVE in '92　恋をしようよ Yeah! Yeah!/ゆめいっぱい
THE SOFTOUCH　ファイナルファイト/ライジングサン/
ヨーロッパ戦線/シューティング68K GAMES
全機種共通システム　O-EDIT＆MODCNV

10月号
特集　DTMへの招待
連載　DōGA CGアニメーション講座/大人のためのX68000
響子 in CGわ～るど/吾輩はX68000である/ショートプロ
マシン語プログラミング/ハード工作/ANOTHER CG WORLD
●試用レポート　X68000用CD-ROMドライブ
LIVE in '92　美少女戦士セーラームーン/笑顔を探して 他
THE SOFTOUCH　ポピュラスII/リーディングカンパニー/
ネクタリス/サークII
全機種共通システム　実践Small-C講座(6)SLENDER HUL

11月号
特集　ゲームマネージメント
連載　DōA CGアニメーション講座/大人のためのX68000
響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW
ハード工作/ANOTHER CG WORLD/Computer Music入門
●新製品紹介　CHART PRO-68K
LIVE in '92　ストリートファイターII/スーパーマリオ 他
THE SOFTOUCH　キャッスルズ/シュートレンジ/
ポピュラスII/サンダーレスキュー
全機種共通システム　実践Small-C講座(7)EDIT

ANOTHER CG WORLD
ザーターさんへおねがい!!
・1等の年末ジャンボ宝くじとその前後賞
・AMIGA4000
・お家
KYOKO
ものを作ってて、アイデアが浮ばないとき、どーしますか…
ビデオを借りまくってひたすら観る…
おいしいものを食べたり好きな音楽を聞いたりする…
ぱくー
いっしんふらんに精神統一をする
ゴーー
へーそーなの?
いんや
あたくしめのばあい…
ぜーんぜんちがうのよーこれが…





今回の
CGデータ

総物体数　507

うちメタボール数　43

光源　2

1280×1024ピクセル

1670万色フルカラーを

4×5ポジで出力

使用ソフトは,

C-TRACE

サイクロン

アヴァン・ポップで仮想空間から逃げ出せ

virtualとreal

「神はrealだ，もし，intと宣言されていないのならば」という英語の冗談があります。ここで，intのほうの意味は，プログラミング言語における型宣言に出てくる「整数型」のことです。一方，realのほうには2通りの意味を持たせており，これこそが，この文を冗談たらしめているといえます。これ以上解説するとヤボといわれそうですが，要するに，文の前半部を読むときには，realは「実在する」という意味なのであり，後半を読むと「実数の」という意味にパッと変わるのです。

プログラミング言語的には「実数の」という意味を持つ単語であるrealは，intと対をなすといえます。一方，「実在する」という意味でのrealと対になりうるのが，virtualといえましょう。このvirtualということばは，単に実在しないということだけではなく，実在するように人間には見えるという意味をも含んでいるところが，面白いところです。

光学的な使い方を例にとればよくわかります。real imageといえば「実像」のことであり，virtual imageといえば「虚像」のことです。理科で習いました。虚像とは，そこにあるように思えるのだが，実はそこに実在するのではなく，そこに見えるように光学的に仮想的な像を結んでいるということです(わかったようなわからないような説明で失礼)。

virtualということばは，計算機関係では比較的よく使われ，仮想という日本語訳がつけられます。仮想アドレス，仮想メモリ，仮想オペレーティングシステム，仮想マシンなどです。これらは，いずれも，プログラム，ユーザー，プロセッサなどに対して，表面的に見える部分の下に階層的にもうひと皮分だけ別の層を用意することを意味します。これにより，都合のよいように表面的には見せることができるようになるわけです。もちろん，複雑さやスピードなどの点からはマイナス要因となりますが。

仮想現実ということば

virtualとrealというこの2つの対となることばを，realのほうを名詞にして結びつけると，実にホットなことばのできあがりです。virtual reality，バーチャルリアリティ，(素直に訳して)仮想現実です。ここで，どうして実在しないものと実在するものが結びつくのか，実在することと実在しないこととの差はなんなのか，などの哲学の認識論の領域に属する話がわいてきそうです。でも，サイバーパンク，ハッカー，出たツイ，SFII'，ギャラ³などのキーワード群を軽々と消化できる(素養のある)読者の方々ですので，ここでは特に寄り道する必要はないでしょう。

でも，ひとつだけ，もの申したいと僕が思うのは，virtual realityということばを日本語にする場合や，virtual reality的な概念や研究のことを呼ぶ場合に，「人工現実感」，よくて「仮想現実感」ということばを使いがちなことに対してです。

このようなことばでは，せっかくもともとのことばが持っている，virtualとrealという対極的な概念を結びつけたことによる面白さが消えてしまっているだけではなく，「人間の五感(あるいは六感)に対して，もし自由にかつ理想的に刺激を加えたならば，その人間にとってはそれが現実そのものになるのだ」というサイバーパンク的あるいは認識論的面白さも失っていると僕には思えるのです。

現実のように見えて面白いんですよというのではなく，ノイズやスピードの点で問題がなければ，人間にとっての現実そのものを本当に変えてしまう可能性があるといったような，根本まで踏み込んだテーマを提起していると思えるのです。

実は同じような疑問を投げかけている人を最近発見しました(文献1)。そこでは，もう少し厳密な議論をしています。artificial realityやvirtual realityということばを使い始めた人の主張，artificial realityということばはアメリカではあまり使われていないという事実，英語の日本語訳としての問題などがあげられています。そしてさらに，このようなことばが使われてきたのは，この分野の第一人者の教授が使用してきたからなのだとしています。

仮想空間に漂う

ある晴れた日の昼下がり，僕は若い衆に連れられて「仮想現実」を体験しに行ったのです。お邪魔した先は，名古屋工業大学機械工学科の藤本研究室です。大学の工学系としてはごく平凡な研究室の中に，Macintoshにつながれた，お値段にして1,000万円を超すという仮想現実装置「MicroCosm」なるものが実在していました。

どのように仮想空間で遊んだかということを説明しましょう。左手に，スキーのゴーグルをごつくしたようなもの(アイフォン)を持って，目にあてがいます。ゴーグルの横から棒が下に突き出ているので，それを持てばいいのです。アイフォンの裏側には液晶テレビのようなものがついており，左右の画面を少しずらしていますので，Macで作り出した画像が立体的に見えます(立体視)。一方，右手には電線(光ファイバ)がはい回っているスキー手袋のようなもの(グローブ)をはめます。システム構成を左図に示しましょう。

これをつければ，あなたも仮想空間を漂うことができるのです。写真を見てください。この人は，実在しない積み木を実在しない空間の中であっちに持っていったり，こっちに持ってきたりしています。アイフ

図　仮想現実システム

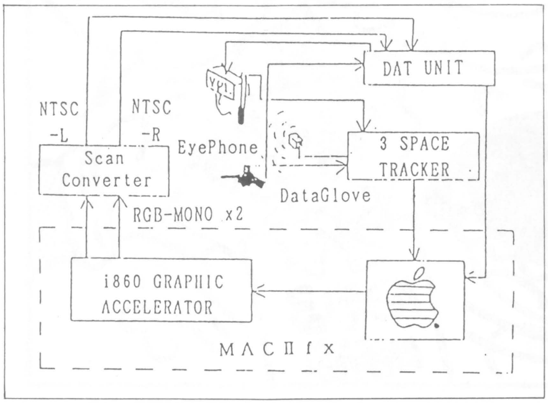
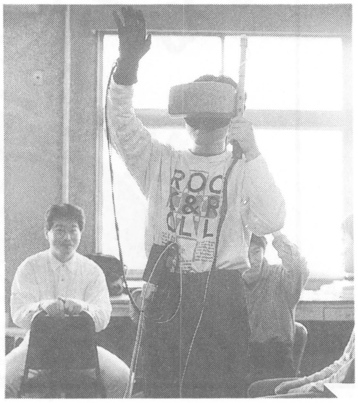


写真　仮想空間をさまよう馬鹿もの



ォンの中には自分の右手も映っており，手や指を動かすと，それをグローブのセンサーが検出して，そのとおりに仮想空間に投影してくれるのです。むろん，座ったり歩いたりすると，背景もちゃんと変化します。

この無邪気で憎めない若者の行動についてもっと正確に記すことにしましょう。彼は，順番待ちしているあとの人が番が回ってきたときに困るようにと，簡単には見つからないような場所に一生懸命にいくつかの積み木を移動しているのです。

ちなみに，見つからない場所とは，たとえば次のようなところです。簡単な順にあげます。

1) 最初に体験を開始する体の位置
あとずさりすれば，体の中からグニュっと登場します。
2) 仮想的な机の下の死角になるところ
しゃがんでのぞきこめば見えます。
3) 背の届かないような高いところ
背の低い人には試練となります。ジャンプするのでしょう。
4) 地面の中
地面もきちんと仮想的にありますので，ヌメッと押し込むとあとの人が探すのはきわめて困難になります。

まあ，そんなこんなで，ただただ遊び，さらにはそれだけでなく，研究室にどんなマシンがころがっているのかとブラブラ探し回ったり，冗談ばかりいったり，とにかく，お騒がせなご一行様だったのでした。

もちろん，接待をしてくれた藤本研究室の八木橋氏は，漂うバカどもをただ見物していたわけではありません。こつこつとアンケートをとりながら，親切に解説をしてくれたりしました。今回彼は「3Dマウス」と,「握ったというジェスチャーで物体をつかむハンド」と,「より人間に近い指先によるジェスチャーで物体をつかむハンド」の，使用感の比較のためのデータをとっているのでした。

アヴァン・ポップ

仮想現実に関する今回見物したような現在進行中の研究では，人体の動きを検出し，それをもとにして計算機内に再構成した世界像を，たとえば人の視覚に対して画像として見せて「現実感」を与えることで，仮想現実なるものを作り出そうとしています。

しかし，もう少し深いレベル，意識とか欲望とか行動とかという我々の生活そのものに近いところにおいては，実際のところ，仮想現実そのものが実現されているのではないかという，一見，妙ちくりんな仮説が，実はありうるのかもしれません。

そして，そのような意味，つまり，仮想現実としての浮世＝現実に生きることを問題意識(アンチテーゼ)としてとらえ，そこからの脱出をうたいあげる文化的なムーブメント，それこそが，ラリィ・マキャフリィの提唱する「アヴァン・ポップ」(文献2，3参照)なのです。

アヴァン・ポップとは，語源的には，ポップで，かつアヴァンギャルド(前衛的)なもののことです。そうはいうものの，もともと，どんなに前衛的で過激なムーブメントであっても，結局は，現に存在する流通メカニズムのおかげで，お客さんはきちんと存在します。したがって，そのメカニズムをきちんと構成，維持することになり，そのムーブメント自体のパワーが失われてしまってきたという自己矛盾があります。

極端な例ですが，ポップアートの巨匠アンディ・ウォーホルなどは，作品自体は前衛的であることには違いないのですが，金が儲かるなら，あるいは有名になれるのならば何でもやるといったミーハーでした。

しかし，1990年代の新しいムーブメントであるアヴァン・ポップは，仮想現実技術をはじめとするさまざまなテクノロジーの進化に基づく我々の生活の革新を源としているところに大きな特徴があります。

1980年代から今日に至る主に情報関係のまったく新しい技術は，我々の世界を大きく変えてしまったという事実がまずあります。現実と仮想世界との逆転ということさえ起き始めているのではということです。

たとえば，ディズニーランドはそれを取り囲む世界より現実的であり，映画や広告の中の人間たちは，現実の人間よりもずっと生き生きしているように見えます。それにくらべ，この現代の浮世は，実は牢獄なのであり，皆が誰かに金を払い続けることにより，その誰かが我々を無為な娯楽の中毒にさせているのだと，マキャフリィはかみつきます。今こそ，W・S・バロウズの「ソフトマシーン」の中のフレーズ「現実スタジオ強襲で宇宙を撮り戻せ」ということばが生まれ変わるというのです。

この仮想現実から抜け出すためには，ポップカルチャーを理解し，愛しており，しかも，それが我々の現実との関係を歪める点のみを憎むような，アヴァン・ポップなアーティスト，ポップカルチャー・テロリストが必要だと主張します。敵は，ポップカルチャーそれ自身ではないのです。

ねこじるうどんってなんですか

アヴァン・ポップは，狭い意味では小説の領域でとらえることもでき，広い意味では音楽や映画なども含めてとらえることもあります。ただし，共通しているのは，さまざまなテクノロジーによって多様になってきた各個人個人のリアリティを表現するためのアヴァンギャルドな主題や隠喩をポップな形で保持しているというところです。

ムーブメントとしてのアヴァン・ポップの面白いところは，仮想現実的な技術によって，作られてしまったこの世の中＝仮想現実を仮想現実的な手法でぶちこわそうとしているところでしょう。

小説の作者としては，マーク・レイナー，ハロルド・ジャフィ，キャシー・アッカー，リック・デマリニスなどがあげられます。僕もすでに読んだり，読んでいる最中のものもいくつかありますが，まあ，興味ある人のお楽しみとでもしましょうか？　ちょっと本連載にはそぐわないので，ここでは触れないでおくことにしましょう。文献2と3は，いろいろな作家の短編集となっているので，とっつきやすいと思います。

文献2によれば，舞の海，松本人志，作家・大槻ケンヂ，ねこじるうどん(これはいったいなんだ？)，ソニック・ユース，プリンスまでもが，アヴァン・ポップなのだそうです。これは極端すぎますね。ここまで広げると，うさんくさくなります。せめて，デヴィッド・リンチやウィリアム・ギブソン程度までなら許せるというものです。

参考文献

1) 下野隆生，“AR，VRは「人工現実」「仮想現実」と訳せ”，日経エレクトロニクス，No.566，pp.226-227，1992
2) ラリィ・マキャフリィ，「アヴァン・ポップ宣言」，SF adventure秋季号，p.3，徳間書店，1992
3) ラリィ・マキャフリィ，「ピンチョン以後のポスト・モダン」，ポストモダン小説：ピンチョン以後の作家たち，pp.248-267，白馬書房，1991

第75回

猫とコンピュータ

肩コリと本棚

Takazawa Kyoko
高沢 恭子

寒い季節は体をちぢめて歩いてしまうので，肩コリがひどくなる人も多いようです。でも，寒がりのはずの猫は肩がコラない？　好奇心旺盛のキョウコさんは，さっそく真相究明(?)にのりだします。

「猫には肩コリがない。だって猫は肩がないから」と誰かに教わった。あんまりおもしろいので，ウソでもホントでもいいと思っていた。でも，肩がないとしたら，肩コリもないのがほんとうだろう。

ところが，アララ。円筒型のマッサージ機の上にあおむけに寝て，うっとりと，首と肩をマッサージしている猫がテレビにうつった。チンチラのシルバーだった。

つぎに飼い主が，柄のついたシャレた肩タタキで，肩を叩いてやっていた。やめると，もっとやってと猫がねだる。

パソコン獣医さん

スイッチを入れてすぐの画面だったのでそこまでのなりゆきはよくわからないが，ともかく，肩をもまれて喜んでいる猫がたしかにいた。

あれはただ刺激的なアソビとして楽しんでいただけなのか。それとも，じつは猫にも肩コリがあるのか。「猫には肩がない」というのは，はたしてほんとうなのか。いろいろ推測してみたいけれど，それにはもうすこし知識がほしい。

夫のパソコン友だちに獣医さんがいる。五十嵐恵子さん。K動物病院の分院の院長さんで，小動物臨床病理研究会の責任者でもある。いまは結婚されて，内田恵子さんになった。

内田先生の勤務先は偶然にもわが家のすぐそばなので，ちょっとお会いできるかしらと電話をしてみたら，「本日はお休みでお宅におられます」とのこと。それではとあきらめていたら，なんと先方からお電話をくださった。

「ネコちゃんのことでしょ？」

「ええ，……」

「すぐわかりましたよ」

「でも，病気じゃないんです。どうしてもご意見をうかがいたいことがあって」

猫には肩コリがないと聞いたこと。テレビで肩のマッサージを喜ぶ猫を見たこと。相反する2つの見聞をお話しした。

「それで……，人間でも肩コリがなくてももんでもらえばキモチがいいということがありますが，猫も，そんなふうに喜んでいたのでしょうか。それとも猫にも頑固な肩コリがあると想像できるでしょうか」

「それはおそらく……」

ここで内田先生はすこし考えて，

「肩コリというのではなく，キモチがいいとか，気にいっているということだと思います。なぜなら，いっぱんに犬には頸椎，腰椎の病気が多いのですが，猫にはそういう故障が少ないとされているんです」

「なぜ，犬に多くて，猫には少ないのでしょう？」

「それは，ひとつは運動性のちがいと，もうひとつは長いあいだの品種改良のちがいにあります」

「運動性とは，犬はつながれていて，猫は自由であるというような……」

「そうではなくて，同じ四つ足の動物でも，犬の仲間であるオオカミやコヨーテの運動性と，猫の仲間であるヒョウなどの運動性というのは，だいぶちがうんです」

「品種改良といいますのは？」

「犬は大きいものと小さいものの差がとても大きいのです。それらの間でいろいろな品種をつくるので，無理も生じるのでしょう。それにくらべたら，猫には，それほど大きな体格の差はないですから，新しい品種をつくっても無理が少ないんです」

肩はあった！

「猫には肩がないから肩がコラないというのはどうでしょう」

「猫にも肩はありますよ」

ヤッパリ！

「犬にも猫にも，肩甲骨と肩関節はあるんです。ただし，犬には鎖骨がなくて，肩と前足とは筋肉でつながっているだけなんです。だから，肩のあたりを輪切りにしたら前足はポロンととれてしまいます。でも，猫には鎖骨も痕跡のようなものですが，ちゃんとあります」

体の組み立ては，犬以上にととのっている猫。それでも，ナデ肩の人を「肩がない」というような人間界の表現に合わせれば，やっぱり「猫は肩がない」。そして，肩コリもどうやらあまりないらしい。

動物には体幹皮筋というものがあって，人間にはない動きをみせるそうだ。牛が背中にとまったハエを，その部分を動かしておいはらうような運動だ。犬などもよく体をさすってもらうと喜ぶが，それは体幹皮筋をはじめ，ヒフの状態がよくなるからだという。

動物病院でも，苦痛をともなう治療であっても，そのあとのぐあいがよいということがわかると，とくに犬などはけっして拒まなくなるそうだ。猫が電動のマッサージ機を喜ぶのも，なにかよい効果があるからだろうと内田先生はおっしゃる。

みじかい時間にいろいろ教えていただいてしまったが，話が終わろうとするとき，

「猫は空を飛ぶんですよ」

また興味をひくような先生のひとこと。

「ムササビ現象ですか」

「高いところからの落下のさいに，じょうずに回転するのは知られてますよね。まあ，ああいったことなのですが，アメリカの研究では，低い階からの落下より8階以上か

らの落下のほうが生存率が高いという結果が出ているんです」

ハイライ・シンドロームと呼ばれる現象の研究があるのだそうだ。近いうちにK動物病院にぜひ出かけてみよう。

フォーマット

トオルのために本棚をひとつふやした。いまある２つのものより，こんどはちょっと大きく上等で，色あいもつくりもなかなか重みがある。

コンピュータならメモリ増設というところ。これでトオルの書籍類はゆうゆうと出し入れできるようになり，さらに本がふえることにも備えられると期待した。

ところがこれが，その大きさに見合うだけの収納力が発揮できるかというと，そうでもない。

原因は仕切られた棚にある。ハードディスクやフロッピーディスクならフォーマットの段階だ。本棚の場合は，仕切りの板を上下にいくらか調節できるが，上がふえれば下はせまくなるのだから，全体には大きな変わりがあるわけではない。

できた棚の高さに合わせ，本はまずサイズ別にそろえられていく。でも，できれば分野別にもおさまってほしい。両方を考えながら，あれこれ置き換えてくふうする。

本のサイズは，じっさいには奥行きもふくめてたくさんの種類があり，思いのほかバラバラなものだ。わずかなところでそろいのサイズからはじきだされる。分野もわけていけばキリがない。ハンパな迷子がたくさん出てきて，なかなかスッキリとはおさまってくれない。

とくに百科事典や全集ものがあると，優先的にスペースを占有するので，全体的な棚の分割はこれに大きな影響をうける。

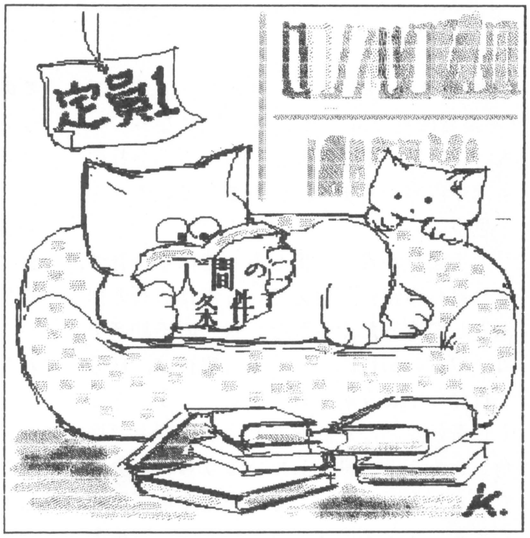
こうして，はじめはたのもしそうに見えたりっぱな本棚でも，その何割かがムダな空間となって，書物の積み残しを出してしまう。

その点，コンピュータならディスク容量いっぱいまで，データのすきまを見つけて有効にファイルの収納ができるだろうと思ったら，これもそうではなかった。

まるいディスクはフォーマットによって仕切りがつくられる。同心円のトラックがひかれ，トラックは中心から放射状に分割されてセクタができる。このセクタの棚の中にデータがならんでいく。

ひとつのセクタの中には一定のバイト数（たとえば256バイト）のメモリが入り，満たされるとつぎのセクタに進入していく。最後のセクタに余りができても空白のまま残り，使われることはない。あたらしいデータは，となり，または別のセクタから記録がはじまる。

こうしてみると，本棚もディスクもひと目で分類がわかり，取り出しやすいゆとりがなくては困るのだろう。書物の展示棚としての本棚も，そのほうが美しい。すきまがあるからといって，タテ，ヨコかまわず詰め込むのは，使いにくいだけだ。



illustration：Kyoko Takazawa

花さくライブラリ

仕切り棚のセクタのおかげで，トオルの新しい本棚はゆとりの空白をつくりながら姿がととのった。いままでの２つの本棚にも，それぞれの分野にあらたな本をむかえる余裕ができた。

16歳の本棚は楽しいものだ。学習机に近いあたりは教科書，参考書で埋まり，ノート類もびっしり。それから先は，16年の道のりをすこしずつのぞかせて，さまざまな本が見知らぬ乗客のように肩をならべている。

３冊の古書『三國志』。森村誠一版『忠臣蔵』。「カルメン」の全楽譜。

星新一，筒井康隆。

芥川。『古典落語』。

コミック『美味しんぼ』『三國志』『キン肉マン』の全巻。『パーマン』『ハットリくん』『Dr.スランプ』。

大判の本が入る棚は音楽誌の洪水。月刊誌や楽譜。米米クラブや，ギタリスト布袋寅泰のバンドスコア。ロックにまじって，バッハ，シューベルトの楽譜集。

となりあわせた本の背中で，タイトルが連想ゲームのようなあそびをやっている。『変身』『モンスター事典』『社会科見学・国会議事堂』『暗黒要塞』。

興味深いのは，書棚がオーディオ部門のライブラリも兼用していることだ。音楽テープ，ビデオテープ，コンパクトディスクが，スライド式の本棚にふしぎにもぴったりと寸法が合って整列している。

アナログの書籍もメカニックな記録物も，同じレベルで相乗りさせて情報源にできるところが，現代っ子だなと思う。

「トオル君の本棚は，生き生きしてるし，よく利用されていて，いいなぁ」

子供時代の新宿の家で，天井までいっぱいにつくりつけられていた父の本棚を，ふと思いうかべた。

長い長い行列の『源氏物語』，重量級の『万葉集』。『カラマーゾフの兄弟』『戦争と平和』『大地』『罪と罰』。

どれもずっとすわりつづけて褐色になった，「蔵書」という名の家具だった。ひと目で作品の長大なことだけはわかったが，手にとるのは気がおもかった。

そのころ，『科学の学校』という岩波書店のグラフ誌があった。いまの『ニュートン』のような科学誌で，子供たちに科学の知識や理解をもたらし，関心を高めさせようというものだった。私と兄は，毎晩これで母の講義をうけさせられたが，私にとってはなんの効き目もなかった。

「せめて，いまのように絵が動いてくれたらねぇ」

IDCソフトの中西秀樹さんから，化学教育ソフトの試作品が送られてきたのは，その数日後だった。

「前作をバージョンアップしたもので，来年１月１日発売予定です。ぜひ，感想をお聞かせください」

と，メッセージがそえられていた。

―以下次号へ―

銀行のBIS基準クリアをめぐって，最近，「不良資産買い取り会社」が話題になっている。この号が出るときは，おそらく会社の名前や内容が決まり，公表されていることと思う。

大手銀行が資金を出しあって作る会社で，それぞれの銀行が抱えている不良債券の担保としてまったく動かない状態の不動産を買い取るのが目的。つまり各銀行は，自分で自分の不良資産を買い取るという，まことに奇妙な話だが，とりあえず帳簿上は換金できること，動かしているうちに処分もできるかもしれないこと，換金を無税償却できることなどのメリットがあり，実施の運びとなったものだ。

この場合，不動産自体が「不良」なのではなく，これを担保にしている債券のほうが「不良」(半年以上利払いが遅れている返済されるべき貸付金など)であり，いわゆる「不良資産」とは意味が違う。

とはいえ，広義に解釈すれば，これも「不良資産」といっていいだろう。「担保不動産つき不良債券」をはじめとして，バブル経済の崩壊後は，この「不良資産」という言葉は大きなキーワードのひとつになっている。

メーカーの場合は「不良在庫」に頭を痛めている。最も身近な例として，古い商品を在庫として保管しているメーカーがそうだ。1世代前のオーディオ製品なら，安値で卸せばまだまだ商品価値はあろうが，2世代以上前のOA機器ともなると，それは期待できない。少なくともPC-9801系などごく一部のメーカーの古いパソコンなど金をもらっても欲しくない，という人がほとんどのはず。しかも帳簿上はけっこうな価格がついたままなのだから，いかにも「不良資産」という表現がマッチする。

メーカーとしては，どこで見切りをつければいいか判断がつかずに倉庫に保管していたり，管理不十分で残ったままになっているものが大量にたまっている。いきおい，どこかの決算の段階で処分しなければならず，悩みの種。

この「資産」「不良資産」という考え方，われわれ一般個人にも，応用するとためになる。経済学でいう「家計」を単位とする理論的な話ではなく，非常に漠然とした話なので，気楽に読み流していただきたい。

われわれ個々にとっての「資産」といえるのは，土地，建物，持ちもの，現金や貯金ということになる。権利関係や大金をかけているペットの扱いなどは難しいところ。

ここで「不良資産」があるとすれば，どのようなものがそうなのか？

1)　相手が行方不明になってしまっている貸付金の債券

2)　貸した土地の借り主がまったく金を払わなくなった場合

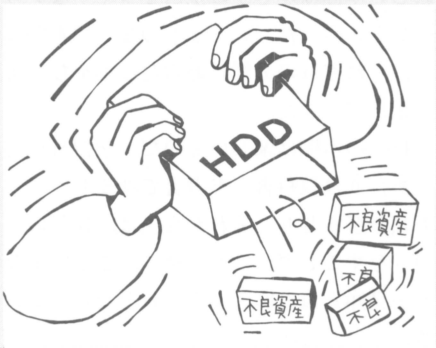
3)　高い価値があると思ったまま持っている物品

こんなところだろうか。さらに3)が派生して，持っていることを忘れてしまっている物品や金融商品なども該当しそう。

1)や2)はさておき，あとは現実に誰もが

X-OVER・NIGHT
(クロスオーバーナイト)

［第29話］
不良資産



TAKAHARA HIDEKI 高原 秀己

体験している問題だ。具体的には押入れの中につっこんであるものや，部屋の隅に置いてある本棚で，ここ数年まったく手を触れておらずホコリまみれになっている段に置いてある本や品物が頭に浮かぶ。

こうしたものは，大半が不要である，と断言していい。なにしろまったく使っていないのであるから，持っていても何の価値もない。しかし，保管してあるということは，後日何かの役に立つだろうということを想定し，その時期においては「資産」として評価した，ということを示している。

文字どおりの個人版不良資産，である。企業の場合は，こうしたものはある時期に損失として計上し，決算上で処分する場合が多い。特別損失として放棄するケースだ。これを個人で行うには，帳簿はとくにつけていないだろうから，「処分」という作業のみが問題となる。

この作業の効果として，処分しないものを洗い出せる点が指摘できる。たとえば机の後ろに落ちて行方不明になっていたローレックスの時計もそうだし，持っていることすら忘れてしまっていた商品券やビール券もそう。もうちょっとブレークダウンすれば，処分前の古着に紛れ込んでいた包んだままの新品のポロシャツもだし，もらったまま押入れにつっこんであった乾物もだ。

とどのつまり，この話，ちゃんとお掃除すると，得することがありますよ，という子供のしつけみたいな話でもある。子供のしつけで掃除させる場合なんかも，最近の子供は勘定高いから，お母さんがヒステリックに「お掃除しなきゃいけないからしなさい！」と怒るよりも，「何を持っているか，毎月調べて報告しなさい」というほうが効果的なのかもしれない。

勘のいい読者の方はすでにお気づきであろうが，何のことはない，ぼくが最近，この不良資産処分を目的としたお掃除を延々と続けている，という実体験に基づいた話なのだ。そもそもの目的が掃除のための掃除ではなく，不良資産処分のための掃除であるから，単純に見栄えをよくして終わり，という日頃の掃除とは根本的に性格が異なる。品物の価値判断をいちいちやって「不良」なら思い切って捨てることにしている。

ちなみに効果であるが，膨大な「不良資産」の処分と，意外に豊富な資産の再確認，さらには意外な発見として，「含み資産」まであったことが確認できた。部屋もすっきりして，快適になった。

そういえば，前号で話題にしたぼくの壊れたハードディスクだが，辛うじて，蓄積情報を失わずにすんだ。VTRが1台買えるほどの大金はかかったものの，データ保護のためにはしかたがない。目下，前号で提案したように，バックアップ用の新しい2台目のハードディスクを買おうと思ってはいるのだが，さすがに修理されて戻ってきてしまうと，「喉元過ぎればなんとやら」で，腰が完全に重くなってしまっている。

とりあえずは，部屋の掃除もひと段落したので，次はハードディスクの中の「資産」を洗い出しに取りかかろうと思っている。

料金受取人払

高輪局承認

1396

差出有効期間
平成6年7月
15日まで

郵便はがき

1 0 8 - 0 0

5 0 7

(受取人)

東京都港区高輪
2-19-13 NS高輪ビル

ソフトバンク株式会社

Oh!X 編集部行

□□□-□□

電話

住所

氏名

年齢

職業・勤務先
学校・学部・学年

●編集部へのメッセージ

今月号の特別企画について

いちばん良かった記事	興味のなかった記事
これから載せてほしい記事内容	本誌以外にお読みのパソコン雑誌

推薦する市販ソフト

ソフト名：

推薦理由：

CGで表現するのが難しいものは何だと思いますか？

あなたの愛機は(所有機種に○印をつけてください)　ない
X1(マニアタイプ,C,D,F,G,twin) X1 turbo(model 10,20,30,40,II,III,Z,ZII,ZIII)
MZ-(80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861)
X68000(初代,ACE,PRO,PROII,EXPERT,EXPERTII,SUPER,XVI,Compact, HD)
その他　MIDI楽器(　)
FD(　基)　TAPE　QD　HD(　MB)　MO　プリンタ(　)

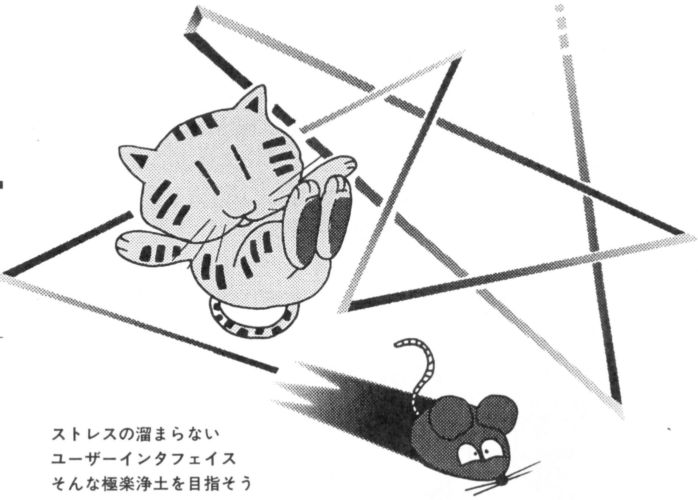
年齢　歳　パソコン歴　年　男・女　プレゼントNo.

吾輩はX68000である

[第18回]

極楽た〜ぼマウス

Izumi Daisuke
泉　大介



ストレスの溜まらない
ユーザーインタフェイス
そんな極楽浄土を目指そう

前回は，ユーザーフレンドリなインタフェイスを目指して吾輩のIOCSコールに用意された，「マウスカーソルのアニメーション機能」を紹介した。X-BASICで作った簡単なマウスカーソルパターン作成ツールも用意しておいたので，さぞやユニークなアニメーションパターンを作成して遊んでいただけたことと思う。さて今回は，吾輩がいったいどのようにしてマウスのデータを取り込んでいるのか，その実際を紹介しておきたい。

◆どうしてそんなにノロイのか

最近，うちの御仁は吾輩のマウスに触るたびにご機嫌斜めである。「画面の端から端までマウスカーソルを動かすのに，マウスマットの端から端までマウスを動かさなければならない」というのがその理由だ。よくマウスを使い始めたばかりのユーザーが，マウスをマットの端まで転がしたのにマウスカーソルが目的の位置まで動かないため，途方にくれているという話を耳にする。マウスを浮かして動かすぶんには，マウスカーソルは動かないということが最初はわからないのだ。もちろん，発売されたばかりの吾輩を購入した古参組である御仁が，こんなところで悩んでいるはずはない。

やはり元凶は，手元の小さな動きに反応してマウスカーソルが画面内を縦横に走り回るようになっているWINDOWSである。もちろんマウスカーソルの移動スピードはユーザーが自由に設定できるのだが，御仁が触るマシンはどういうわけだか最高速に設定されているものが少なくないらしい。締め切りが過ぎても原稿を書かず，毎度のことながらついには缶詰になって原稿を書くはめに陥る御仁は，その最中にもしばしば「息抜き」と称する現実逃避にWINDOWSのカードゲームで遊んでいる。WINDOWSそのものは「なんだかなぁ」とこぼしている御仁だが，このカードゲームはかなり気に入ったようだ。現実逃避が昂じて，原稿の合間に遊んでいるんだか遊びの合間に原稿を書いているんだかわからなくなった頃には，すっかりこの「高速マウス」に慣れ切ってしまっていたという顛末。最近入れ込んでいるMacintoshのマウス移動スピードも，これに合わせるかのように最高速に設定され，スピード固定の吾輩は，いささか肩身の狭い思いをしているのである。

とはいえ，吾輩のマウスにも自慢はある。諸兄もよくご存じのとおり，マウスの移動スピードに応じて加速度的にマウスカーソルの移動スピードが増加する「加速度マウス」になっていることである。ブンと振りゃアッという間にマウスカーソルは画面の端まで移動するし，普通に動かしているぶんにはマウスカーソルは1ドット1ドット動きを忠実にトレースする。ドット絵師にはもってこいのこの性格も，アプリケーションを使う身になってみれば少々面倒なもの。正確に1ドットをトレースできなくてもいいから，もっと機敏な動きでメニューを選択できるようになれば，ということらしい。

◆マウスの生データを読む

そこで御仁は，吾輩のマウスカーソルの移動速度を高速化する「ターボチャージャ」の作成を始めた。その話に入る前に，まずマウスの生データを紹介しておこう。

以前，吾輩がどうやってジョイスティックのデータを扱っているのかをご覧に入れたことを覚えておいでだろうか。アドレス$E9A001_H$には，ポートAにつながれたジョイスティックのデータが常にレポートされていた。諸兄は$E9A001_H$のデータを取り出すだけで，ジョイスティックがどちらの方向に倒されているのか，A,Bどちらのボタンが押されているのかを簡単に知ることができた。

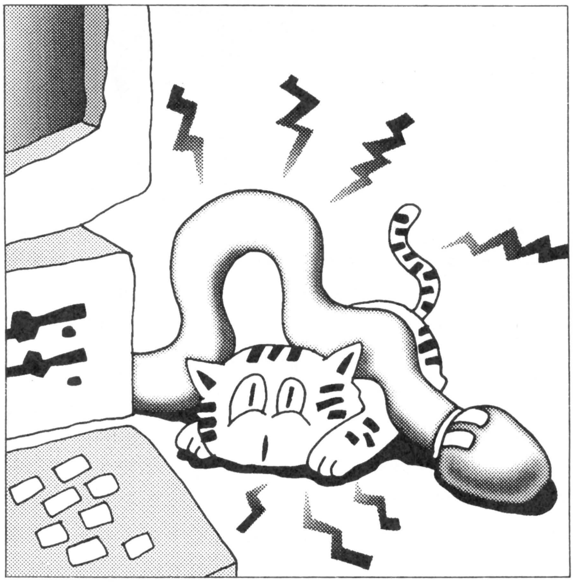
マウスのデータもこのように手軽に扱うことができればそれにこしたことはないのだが，残念ながらそうはいかない。ジョイスティックは「上に倒された，倒されていない」「Aボタンが押された，押されていない」といった，上下左右ABの6つの情報を伝えるだけでいいのに，そのケーブルはあの太さである。左右ボタンの情報に加え，xy方向の移動量をそれぞれ8ビットで伝えなければならないマウスを同様の手法でつないだとき，つまりメモリの特定のアドレスから3バイトに，常にマウスのデータがレポートされるような繋ぎ方にしたときのマウスケーブルの太さは推して知るべし。諸兄も鉛筆のようなケーブルを引きずったマウスを使いたいとは思わないことだろう。

そこで，マウスは1本の信号線のON/OFFを使い，モ

ールス信号のように情報を吾輩に送ってよこすようになっている。これならマウスにデータをよこすように要求する線(MSCTRL)とデータ伝達の線(MSDATA)の2本があればいい。吾輩がマウスにデータをよこせとMSCTRLに信号を出せば，マウスはMSDATAをトンツー・トンツーとやってデータを送ってくるというわけである。

マウスから送られてくる実際のデータだが，これは図2のような形式になっている。本当にモールス信号を使ったのではデータ転送に要する時間が馬鹿にならないので，1/4800秒間MSDATAが0なら0，1なら1が送られてきていると処理するようになっている。つまり，1秒間に4800個の0/1を送ることができるわけだ。これをデータ転送速度4800bps(bits per second)という。また，データの先頭と最後がわかりやすいように，先頭には0を1個，最後には1を2個付加するようになっている。データは1バイト(8ビット)単位で送信されるようになっており，「データ長8ビット，スタートビット1，ストップビット2」となる。まるでRS-232Cを使うときの通信パラメータのようだが，事実そのとおり，マウスからのデータの受け取りはパソコン通信のミニチュア版のようなものである。

図1　太いケーブルの付いたマウス



マウスから送られてくるデータを受け取る方法は実に簡単だ。1/4800秒ごとにMSDATAが1になっているか0になっているかをチェックすればいい。データの狭間に当たるとやばいので，1/9600秒程度の頻度でチェックすれば完璧である。ただしこの方法には問題がひとつある。常にMSDATAを見張っていなければならないため，吾輩はほかになにもできなくなってしまうのだ。

そこで吾輩には，RS-232Cの制御およびマウスデータの受信のためにZ8530SCCというチップが搭載されている。SCCというのはシリアル通信コントローラ(Serial Communication Controller)の略で，1本のラインの1/0でデータをやりとりするシリアル通信処理を，自動的にこなすチップである。具体的には，SCCに1バイトのデータをセットすれば，それが自動的に1/0の並びとして送信され，1/0の並びが入ってくれば，それを自動的に1バイトのデータにまとめてくれる機能をもっている。さらにデータを受信すると「データが入ったよ」と教えてくれる機能まであるので，吾輩はそれまでの間自由に諸兄が要求する処理を行うことができるのである。

マウスからデータが届いたことを教えられた吾輩はSCCからデータを受け取り，それをいったんメモリに保存しておく。図3はそうしてメモリに保存した，マウスから届いた生のデータである。この生データが保存されるアドレスは，現在までのところすべてのX68000で共通になっている。諸兄もマウスを動かしながらアドレス930_Hをダンプしてデータを確認してみていただきたい。

データはSCCによって1バイトずつ受信され，そのたびに吾輩は作業を中断してデータを受け取る。そしてステータス，x方向移動量，y方向移動量の3つのデータが揃ったら，そのデータをマウスデータを保持しているワークエリアにコピーし，演算を加え，マウスカーソル

図2　マウスから送られてくるデータの構造

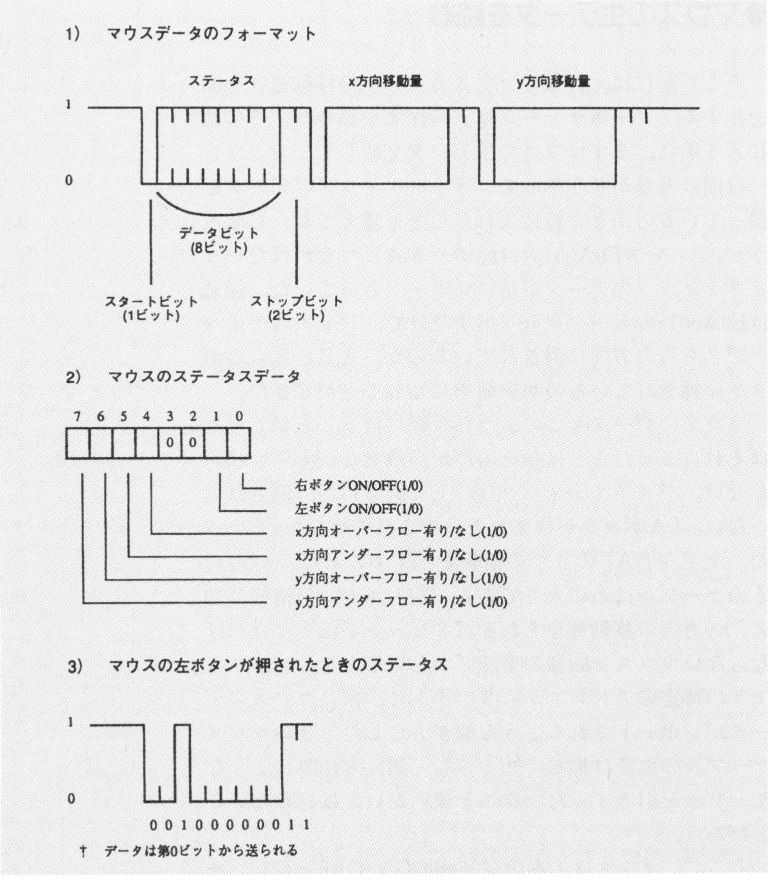


図3　メモリに保存されたマウスデータ

1)　マウスに手を触れない場合

```
-ds 930 932
00000930  00 00 00
```

2)　マウスの左ボタンを押しながら右に動かした場合

```
-ds 930 932
00000930  02 0F 03
```

y方向移動量
x方向移動量
ステータス

の座標を更新し，必要ならアニメーションを行うためにマウスカーソルを変更するのである。

ところで吾輩のマウスは，いつもいつもデータを垂れ流しているわけではない。最初に述べたように，吾輩がMSCTRLを使って「データをよこせ」と指示したときにのみ３バイトのデータを送ってくるのである。では，どんなタイミングで吾輩はMSCTRLを操作するのか，た～ぼマウスの研究を始めた御仁がまず目をつけたのはここである。

◆マウスとタイマC割り込み

吾輩が時計を内蔵していることはすでにお話ししたが，吾輩が内蔵しているタイマは，年月日を管理し，時分秒をカウントしているこのタイマだけではない。グラフィックを復活させるときにMFP(Multi Function Peripheral)に内蔵されている，GPIPのレジスタを操作したことを覚えていらっしゃることと思うが，このMFPの中にも４つのタイマが内蔵されているのである。４つのタイマにはタイマＡ～タイマＤの名前がつけられており，一定時間ごとに「時間だよ」とMC68000に通知してくるようになっている。この中の「タイマＣ」からの通知を合図に，吾輩はMSCTRLを操作して「データをよこせ」とやるようになっている。

タイマＣからの時間通知は「タイマＣ割り込み」と呼ばれている。この通知があるとMC68000はそれまで実行していたプログラムを一時的に中断し，あらかじめ用意されている別のプログラムを実行する。このプログラムが入っているアドレスは，アドレス114_Hに保存されている。吾輩(初代)の場合，ここに収められているアドレスは$FF133C_H$である。114_Hにどんなアドレスが収められているのかはROMのバージョンによって違うので各個に確認してみていただきたい。

さて図４は$FF133C_H$以降をデバッガで逆アセンブルしたリストである。機種によって多少内容は異なっているかもしれないが，やっていることは同様なので参考にしていただきたい。ここにはa0.lにアドレスをセットしてサブルーチンを呼び出すという処理が４度にわたって記述されている。で，そのサブルーチンはというと図4-2である。図4-3はこれらのサブルーチンで参照されるデータで，リストを読むと図4-4のようにデータが利用されていることがおわかりだろう。たとえば$9B2_H$からの８バイトはタイマＣ割り込みがかかるたびに$FF1424_H$が呼び出されることを示しているし，$9BA_H$からの８バイトはタイマ割り込みが32_H回かかると$6F088_H$が呼び出されることを示している。ちなみにこの$9BA_H$のデータはカーソルの点滅スピードを決めるもので，試しにこれを半分の19_Hにしてみるとカーソル点滅スピードが２倍になる。デバッガで，

　　－mes 9ba

とやり，

　　0009ba　　32：■

と表示されたところで，

　　0009ba　　32：19■

とタイプしてリターンキーを押せばOKである。

御仁の最初の戦略は，タイマ割り込みに従って吾輩がMSCTRLを操作しているなら，割り込みをもっと頻繁にかけてやれ，ということだった。単位時間当たりにマウスから送られてくるデータが２倍になれば，マウスカーソルの移動量も２倍になると考えたのである。マウスにデータ転送要求を出す処理を司っているのは，先ほども見た図4-3の$9B2_H$からの８バイトである。こちらはタイマＣ割り込みがかかるたびにマウスにデータ転送要求を出すようになっているため，カーソル点滅のように簡単にスピードアップするわけにはいかない。

そこで御仁は，タイマＣ割り込みそのものの頻度をアップすることを考えた。具体的な方法はまたいずれお話しするとして，ここではその試みも失敗したことをお伝えしておこう。勘のいい諸兄ならその理由はおわかりだろう。マウスカーソルの移動量はマウスの移動量によって決まるのであって，単位時間当たりのデータ転送量によって決まるのではない。マウスが送ってくるxy方向の移動量は，前回吾輩がデータ転送要求を出してからマウスがどれだけ動いたかである。それを一気に送ろうと，細切れに送ろうと，送られてくるデータは実際にマウスが移動した距離であることに変わりはないのである。

◆SCCのチャンネルB割り込み

SCCはデータが届くとそれを１バイトにまとめ，MC

図4　タイマC割り込みの処理

1)　割り込み処理プログラム

```
-l ff133c ff1360
 00FF133C   move.l  A0,-(A7)
 00FF133E   lea     $000009B4,A0        A0.lにアドレスをセットして
 00FF1344   bsr.s   $00FF1362           FF1362Hを呼び出す
 00FF1346   lea     $000009BC,A0        以下この繰り返し
 00FF134C   bsr.s   $00FF1362
 00FF134E   lea     $000009C4,A0
 00FF1354   bsr.s   $00FF1362
 00FF1356   lea     $000009CC,A0
 00FF135C   bsr.s   $00FF1362
 00FF135E   movea.l (A7)+,A0
 00FF1360   rte
```

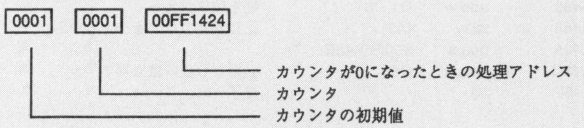
2)　FF1362Hのサブルーチンの内容

```
-l ff1362 ff1370
 00FF1362   subq.w  #1,(A0)             アドレスA0に入っているデータから1を引き
 00FF1364   bne.s   $00FF1370           0でなければFF1370Hへ
 00FF1366   move.w  $FFFE(A0),(A0)      0ならアドレスA0-2のデータをA0へコピーして
 00FF136A   movea.l $0002(A0),A0        アドレスA0+2に入っているデータをA0に取り出す
 00FF136E   jsr     (A0)                そのアドレスを呼び出して
 00FF1370   rts                         終了
```

3)　9B4H-2以降に入っているデータ

```
-d 9b2 9d1
000009B2  0001 0001 00FF 1424 0032 0005 0006 F088
000009C2  00C8 0067 00FF 1498 1770 08FB 00FF 14BA
```

4)　データの内容



68000にデータが到着した旨を伝えると説明した。これがSCC割り込みである。吾輩に内蔵されているSCCは2つのチャンネルをもっており，チャンネルAはRS-232Cの通信に，チャンネルBはマウスデータの受信に使用されている。マウスからデータが届いたときにかかるのはチャンネルB割り込みである。

タイマC割り込みのときと同様に，SCCのチャンネルB割り込みがかかったときにもMC68000はそれまでのプログラムの実行を中断し，あらかじめ用意されている別のプログラムを実行する。このプログラムが収められているアドレスは150$_H$に入っている。吾輩の場合，150$_H$にはアドレスFF1502$_H$がセットされているが，これもROMのバージョンによって異なっているので自分のマシンで確認していただきたい。そしてFF1502$_H$には，図5のようなプログラムが収められている。

SCCにデータが届いて割り込みがかかると，吾輩はまずステータスレジスタ(SR)を操作してそれ以上割り込みがかからないようにする。そしてSCCのチャンネルBに届いたデータをd0.wに取り出す。続いて取り出したデータをメモリに保存するのだが，リストを見ればおわかりのように，データを保存するアドレスを92C$_H$から取り出すという回りくどい方法になっている。マウスから送られてくるデータは，ステータス，x方向移動量，y方向移動量の3バイトである。つまり，タイマC割り込みで吾輩がマウスにデータ転送要求を出すと，SCC割り込みはこれらのデータが到着するたびに合計3回発生することになる。データ保存アドレスを92C$_H$に格納しFF151E$_H$でカウンタをひとつ小さくしているのは，この3度の割り込みに対応するためである。

図5では省略してあるが，マウスのデータが3バイト揃ったあとには，マウスカーソルのアニメーション，座標の更新といった，実にさまざまな処理が待ちかまえている。その中から諸兄が最も興味をもたれるであろう加速度マウスを実現している部分を紹介しておこう。

諸兄の中には吾輩のマウスが加速度データを送ってくるのではないかと思っている方がいらっしゃるかもしれないが，実際は図6のようなプログラムによって実現されている。諸兄のマシンでこのプログラムがどこにあるのかを提示できればそれぞれ確認していただけるのだが，サブルーチンのサブルーチンのサブルーチンといったような実に奥まったところに入っているため，「このようなプログラムが実行されているのだな」ということで勘弁していただきたい。努力と気力と根性の人なら，図5のプログラムを追いかけていけば発見できることだろう。

加速度マウスのプログラムは，マウスの移動量をワード長に直したデータがd0にセットされて呼び出されるようになっている。複雑な計算が行われているが，煎じ詰めれば$d0^2/8+d0^2/32$という計算が行われているにすぎない。ここで注意しておきたいのは，FFA478$_H$でd1を8で割っている部分である。ここで余りが切り捨てられるため，移動量が0～±15ドットならd1にセットされるデータは1となり，計算結果にはほとんど影響を与えないことを意味している。加速度マウスが本領を発揮するのは，移動量が16以上のときなのである。冒頭でマウスを動かしながら930$_H$をダンプしてみるという実験を行っていただいたが，マウスをかなり速く動かさなければ移動データが10$_H$より大きくならないことに気づかれたと思う。加速度マウスの恩恵を受けるには，本当にマウスを加速度的に動かさなければならないのである。

御仁の2番目の戦略は，それならマウスから送られてくるデータを大きくしてやれ，というものであった。マウスの移動量データは，「前回データを送ってからどれだけ移動したか」である。データを送る頻度が下がれば，送られてくる移動量はより大きくなる。具体的には，吾輩がマウスにデータ転送要求を発する頻度を下げればいい。御仁はデバッガを使って，タイマC割り込みがかかるたびにマウスにデータ転送要求を出すよう指示している前述の9B2$_H$のデータを書き換えてしまった。

図5　SCCチャンネルB割り込みの処理

1)　割り込み処理プログラム

```
-l ff1502 ff1526
 00FF1502   ori.w  #$0700,SR              割り込み許可レベルを7にする
 00FF1506   movem.l D0-D1/A0-A1,-(A7)     レジスタを保存
 00FF150A   move.w  $00E98002,D0          受信データを取り出す
 00FF1510   movea.l $0000092C,A0          データ保存アドレスをA0にセット
 00FF1516   move.b  D0,(A0)+              取り出したデータを保存
 00FF1518   move.l  A0,$0000092C          A0を保存
 00FF151E   subq.w  #1,$0000092A          カウンタから1を引く
 00FF1524   bne.s   $00FF158E             ゼロでなければFF158EHへ
...........................................................
 00FF1526   lea     $00000930,A1          A1に930Hをセット
 00FF152C   move.l  A1,$0000092C          それを92CHに保存
 00FF1532   move.w  #$0003,$0000092A      受信データ数3を92AHにセット
            :                                              ↓
            :                                   データ3バイト
                                                受信時の処理
-l ff158e ff159a
 00FF158E   movem.l (A7)+,D0-D1/A0-A1     レジスタを取り出して
 00FF1592   move.w  #$0038,$00E98000      SCCの最上位の割り込みをクリア
 00FF159A   rte                           復帰
```

† XVIでは2行目にbset.b #5,$933が追加されている

2)　SCCのアドレス

E98000$_H$	0	チャンネルBコマンド
E98002$_H$	0	チャンネルBデータ
E98004$_H$	0	チャンネルAコマンド
E98006$_H$	0	チャンネルAデータ

図6　加速度マウスを実現するプログラム

```
-l ffa46c ffa48e
 00FFA46C   clr.w    -(A7)        正負判定フラグ用
 00FFA46E   tst.w    D0           移動量データをチェック
 00FFA470   bgt.s    $00FFA476
 00FFA472   addq.w   #1,(A7)      負の数ならフラグを立て
 00FFA474   neg.w    D0           移動量データを正の数にする
 00FFA476   move.w   D0,D1        移動量データをD1にコピーし
 00FFA478   lsr.w    #3,D1        それを8で割る
 00FFA47A   bne.s    $00FFA480
 00FFA47C   move.w   #$0001,D1    商が0なら1をセット
 00FFA480   mulu     D1,D0        商とD0を掛け合わせる
 00FFA482   move.w   D0,D1        積をD1にコピーし
 00FFA484   lsr.w    #2,D1        4で割る
 00FFA486   add.w    D1,D0        商を積に加える
 00FFA488   tst.w    (A7)+        正負判定フラグをチェックし
 00FFA48A   beq.s    $00FFA48E
 00FFA48C   neg.w    D0           必要なら負の数に戻す
 00FFA48E   rts                   終了
```

9B2Hのデータを1から2に書き換えたくらいでは目に見えるほどの速度の向上はなかった。3でまずまず。御仁としては4にしたときのスピードが気に入ったようだ。が，データを3にしたあたりから弊害が目立ち始める。御仁は，SCCチャンネルB割り込みでマウスカーソルの書き換えが行われていることをすっかり忘れてしまっていたのである。データ転送を要求する頻度が低下すれば，マウスからデータが送られてくる頻度も低下する。つまり，マウスカーソル書き換えの頻度も低下し，マウスを移動させると始点から終点へジャンプしてくるようになってしまったのだ。

◆た〜ぼマウスのプログラム

御仁が最後にたどり着いた戦略は，SCCチャンネルB割り込みを横取りし，マウスから送られてきたデータに手を加えてやろうというものである。図5と同様のプログラムを自分で作り，データを3バイト受け取ったらx方向移動量，y方向移動量を操作して実際に転送されてきたデータより大きくしようというのだ。御仁の作ったプログラムをご覧に入れよう。図7である。

冒頭の200000H〜200016HはSCCチャンネルB割り込みの処理プログラムを変更する部分である。自分で150Hの内容を直接書き換えてもいいのだが，スーパーバイザモードに移るのが面倒だったらしい。書き換えている最中に割り込みが発生することもありうるので，ここは素直にDOSコールを使っておこうというところか。SCCチャンネルB割り込みには，150Hと154Hの2カ所に処理プログラムのアドレスが格納されているので，これを両方とも書き換えている。

処理プログラムは200018Hから始まっている。ご覧になるとおわかりのように，冒頭部分は図5のROM内プログラムとほとんど変わらない。20003AHの分岐条件を書き直して，データが3バイト揃っていないときの処理を補っただけである。なお，X68000 XVIでは20001CHに，

```
    bset.b  #5,$933
```

が挿入されており，これに従って，

```
    check_x:    .z0+$52
    setnewx:    .z0+$64
    check_y:    .z0+$6c
    setnewy:    .z0+$7e
    cont:       .z0+$86
```

とアドレスがズレるので注意していただきたい。

check_x以降はデータが3バイト揃ったときの処理で，御仁がマウスを高速化するために追加した部分である。マウスの移動量が3より大きいか−3より小さいときには，データを2倍にするようになっている。データの変更が終了すれば，ROM内のプログラムの続きの部分，図5でいうなら，

```
    00FF1526      lea    $00000930,a1
```

へとジャンプして残りの処理を行わせている。

プログラムを実行してみて御仁はいたくご機嫌である。ゆっくりとマウスを動かせば1ドットずつトレースすることができ，ちょっと元気よく動かせばスパッとマウスカーソルは目的の位置へ動いてくれる。残る問題は，メモリ上にデバッガで作ったプログラムは，アプリケーションを実行すると消されてしまうかもしれない，という点だけである。これを防ぐためには，た〜ぼマウスのプログラムで使用している旨をHuman68kに通知しておかなければならない。残念ながら誌面も尽きてきたので，これは次回の話題とすることにしよう。まずは極楽た〜ぼマウスをご賞味あれ。

図7　た〜ぼマウスのプログラム

```
-z0=200000
-an .z0

    †_exit              equ       $ff00
    †_intvcs            equ       $ff25

    *
    *        Turbo Mouse
    *

    †        pea       get_ms_data          get_ms_dataを
00200000     pea       .z0+$18
00200006     move.w    #$54,-(sp)           割り込みベクトル54Hの
    †        dc.w      _intvcs
0020000A     _intvcs                        処理アドレスとして登録する
0020000C     addq.l    #2,sp
0020000E     move.w    #$55,-(sp)           割り込みベクトル55Hも同様に
    †        dc.w      _intvcs
00200012     _intvcs                        登録する
00200014     addq.l    #6,sp
    †        dc.w      _exit
00200016     _exit                          終了

    *
    *        差し替えプログラム
    *

    get_ms_data:
00200018     ori.w     #$0700,SR            ┐
0020001C     movem.l   D0-D1/A0-A1,-(A7)    │
00200020     move.w    $00E98002,D0         │
00200026     movea.l   $0000092C,A0         │ 元のプログラムの焼き直し
0020002C     move.b    D0,(A0)+             │ XVIユーザはbset.b #5.$933を挿
0020002E     move.l    A0,$0000092C         │ 入するのを忘れずに
00200034     subq.w    #1,$0000092A         │ check_x以降のラベルのアドレス
    †        beq.s     check_x              │ もそれにあわせてズレるので注
0020003A     beq.s     .z0+$4a              │ 意
0020003C     movem.l   (A7)+,D0-D1/A0-A1    │
00200040     move.w    #$0038,$00E98000     │
00200048     rte                            ┘

    check_x:
0020004A     move.b    $931,d0              x方向移動量を取り出す
00200050     cmpi.b    #$fd,d0              それが-3よりも
    †        blt.b     setnewx              小さければsetnewxへ
00200054     blt.b     .z0+$5c
00200056     cmpi.b    #3,d0                それが
    †        ble.b     check_y              3以下だったらcheck_yへ
0020005A     ble.b     .z0+$64
    setnewx:
0020005C     add.b     d0,d0                x方向移動量を2倍して
0020005E     move.b    d0,$931              元に戻す

    check_y:
00200064     move.b    $932,d0              ┐
0020006A     cmpi.b    #$fd,d0              │
    †        blt.b     setnewy              │
0020006E     blt.b     .z0+$76              │
00200070     cmpi.b    #3,d0                │ y方向移動量も同様に処理
    †        ble.b     cont                 │
00200074     ble.b     .z0+$7e              │
    setnewy:                                │
00200076     add.b     d0,d0                │
00200078     move.b    d0,$932              ┘

    cont:
0020007E     jmp       $ff1526              ROMのプログラムへ戻す
```

illustration : T. Takahashi

マシン語カクテル
in Z80's Bar
第37回

今月も快調にボケをかますようこちゃんの相手をするのはやっぱり光君。長老もマスターもみんな元気だね。それにしても……いいのか光君，そんなことではいまに尻に敷かれるぞ。というわけで，ちょっとした拡張ユーティリティのお話です。

ユーティリティがほしい

Kaneko Shunichi　金子 俊一

カランコロ～ン♪
源光（以下光）：こんにちは。
ようこ（以下Yo）：いらっしゃいませ。
光：あれ？　ようこさん，情報処理試験はどうしたの？
Yo：それがね……。
長老(以下老)：どうしたんじゃ，真っ赤な顔して。酔っ払ったんか。
光：まさか寝坊でもしたとか。
マスター(以下M)：ちょっと手違いがあったんですよ。
老：試験日を間違えたとか。
光：試験を申し込み損ねていた。
Yo：ピンポ～ン。
光：「ピンポ～ン」じゃないでしょうが。
Yo：ごめんなさいね。あんなにCASL教えてもらったのにね。
光：しようがないですね，4月の試験ではがんばってくださいね。
Yo：はい。
老：これにて一件落着ってか。
光：それではまた来月。
M：ってそんなわけないでしょ。

あなたのワガママ聞かせてください

光：なんだ，今月はなんにもやらずにすむかと思ったのに。
M：そうは問屋が卸しませんよ。
老：ああ，マラソンの。
Yo：宗兄弟のこと？
老：ほっほっほ。
光：それで，今月はなにをやらされるんですか？
Yo：ユーティリティがほしい。
光：へ？
Yo：なにか実用的なものがほしい。
光：そりゃ，ユーティリティっていえば実用的ですからね。
老：どんなものでもいいんかいのう。
Yo：う～ん。
光：それじゃ，最近で不便だと感じたことは？
Yo：え～と，エレベータのボタンが全部押されていて，止まるごとにいちいち閉めるのボタンを押したこと
光：ほかには？
Yo：う～ん。近くのコンビニが12時半までしか開いてなくて，遠くまで買い出しに行ったこと。
光：そんなのばっかり。
Yo：不便だったのよ。
光：エレベータやコンビニはプログラムでは直しようがないですからね。
老：そんなことはわからんぞ。コンビニのネットワークに忍び込んで，ちょこちょこっと営業時間を変更させるとか。
M：エレベータのプログラムを書き直してキャンセルを可能にするとか。
光：それを誌面に載せるとか。
M：やっぱりだめ。
Yo：なんのユーティリティを作るかって意味で聞いていたのね。
光：当たり前です。
Yo：それじゃあX1用にハードディスクを作ってほしいな。
老：ほほう。ちっとは真面目になってきおったわい。

もっと速くアクセスしたい

光：あのさ，ハードディスクってユーティリティでできるもんじゃないよ。
Yo：だってハードディスクの味を知ってしまったんですもの。
老：なに～っ！　おぬしもオトナになってしまったんか？
M：ハードディスクってそういうものだったんですか？
老：そうじゃ。そもそも古来からニッポンの女子は……。
光：（無視）話を続けましょうか。
Yo：やっぱりテープより速くても，フロッピーディスクの限界を感じてしまったわけよ。
M：ありがちですね。
老：三歩下がって夫の影を踏まず……。
光：（無視）それでハードディスクがほしいと。
Yo：うん。
老：朝は誰よりも早く起き……。
M：（無視）でも，RAMディスクのほうがハードディスクより速いよね。
光：そうですね。物理的にヘッドが動いたり磁気円盤が回転するよりは，電気的にアクセスしたほうが断然速いですからね。

仕様を決めよう

光：ようこちゃんってふだんはなにを使ってます？
Yo：やっぱり春麗かな。ときどきガイル少佐も使うけどね。
M：ボケますねえ。
光：そうじゃなくって。
Yo：えっとシャンプーはスーパーマイルドでしょ，ハミガキ粉はアクアフレッシュ3，それから……。
M：ボケたおしますねえ。
光：そうじゃなくって，プログラミングするときなんかに使うツールですよ。
Yo：あぁ，エディタでしょ，アセンブラでしょ，あとはデバッガくらいかな。
光：普通はそんなもんですよね。
Yo：光君はほかにも使うの？
光：プリンタに打ち出しをやる関係で，「MACINTO-C」を使いますね。あとはディスクエディタの「DREAM」を使うかな。
Yo：ふうん。
光：そこで，それらのツールをRAMディ

スクに放り込んじゃえばいいわけだ。
Yo：便利かもしれない。
M：でもRAMディスク取れるほどメモリは余ってないですよ。
光：ふっふっふ。特殊ワークエリアがあるじゃないですか。
M：でもツールを５つも置くほど余ってないんじゃないかな。
光：だったら使ってないRAMを使いましょう。
M：そんなのあるの？
光：X1turboにはグラフィックRAMが96Kバイトも搭載されてるんですよ。
M：“SWORD”では48Kバイトしか使ってないか。
光：片側のバンクが丸々遊んでいますからね。それを使いましょう。

DMAを使ってみよう

光：それじゃ今月はX1turbo用ということで勘弁してください。
M：まあいいでしょう。
光：それではさっそくカチャカチャ……できた。
Yo：まるでお料理番組のようなプログラミングね。
M：ここでデバッガを使って30分間デバッグします，ってか。
光：今回はDMAをふんだんに使ってみました。
Yo：デオキシリボ核酸？
光：それはDNA。DMAはダイレクト・メモリ・アクセスですよ。
Yo：どういう意味なの？
光：CPUを介さずにメモリにアクセスできるって意味じゃないかな。
Yo：なにが便利なの？
光：大量のデータを転送したりするのが速いんだ。
M：今回のテーマに合わせてあるわけですね。
光：ええ。こんな便利なものを使わない手はないですからね。
Yo：実際にはどこで使ってるの？
光：えっと，初期設定のときにメインメモリ上のプログラムをグラフィックRAMに送り込むところとか，プログラムの呼び出しなんかにも使ってるよ。
Yo：DMAの使い方が知りたい。
光：えっとですね，まずDMAを初期化して，データをセットして，動けって命令するだけ。
Yo：わかんない。
光：とりあえず，リストの276行からのデータを説明しておくから，自分でプログラムと見比べてみてね（表１）。もっと詳しい説明がほしかったら「Z80ファミリ・ハンドブック」（CQ出版，額田忠之著）を読むといいですよ。
M：ついでに解説もやってくださいな。
光：はいはい。まず，プログラムはおおまかに３つの部分に分けられているんですよ。
M：ほう。
光：ひとつは初期設定で，あとの２つが今回のメインルーチンになるんです。
M：プログラムはF800$_H$番地から生成するようだから，この部分を壊してはいけないんだね。
光：そうです。一応，#MEMAXのワークにはF7FF$_H$番地と設定しておきましたけどね。X1turboだと，ROMBIOSのワークエリアと重なっちゃうから使用できないんですよ。初期設定ではBIOSのコントロールコードの管理をしてるところをいじくってます。
Yo：ってことは？
光：今回のプログラムはコントロールコードで使うわけだ。
Yo：ふ〜ん？

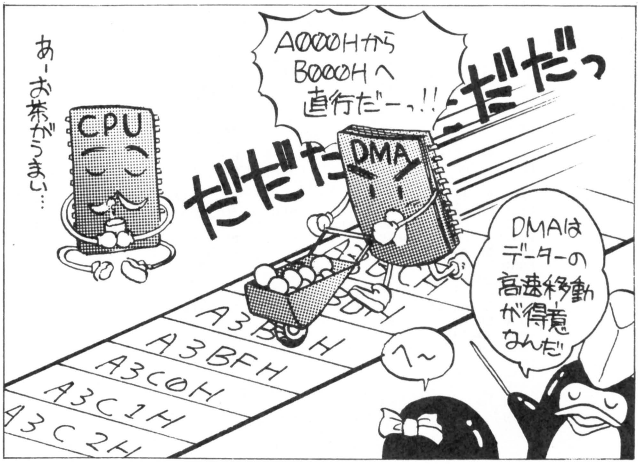
使い方マニュアル

M：使い方の解説もしていただきましょうかね。
光：まずプログラムなんですが，X1用のS-OS “SWORD” で動作します。X1turbo用のS-OSシステムではないんで気をつけてください。
M：それでも動かすマシンは，X1turbo専用なんですよね。
光：そうです。で，“^Q”を押すと数字を聞いてきますから，１〜５で登録したプログラムが立ち上がります。
Yo：数字だけで入力するのね。どれに対応するのか忘れちゃうわ。
光：そんなこともあるかと思って，“M”でメニューが出てきますよ。
M：キャンセルは“０”ですね。
光：ええ。
M：読み込み専門のRAMディスクのようですね。
光：FATとか作って，ちゃんとやってもよかったんですけどね。使い心地がイマイチなんですよ。本当にアクセスが速いだけなんで，自分でメモリにロードして実行しなけりゃならないから。
M：それでこういう形式にしたんだ。
光：MZ-2500にあったアルゴキーみたいなもんですね。
M：確かにコントロールキー＋αでアセンブラとエディタをいったりきたりできれば楽かもしれないな。
Yo：ねえ，登録はどうすればいいの？
光：えっと，“^Q” を押したあとで “E” でエディットできますよ。
M：ここで登録したソフトはこのままで動くんですか？
光：動かないんですよ。
Yo：へっ？

表１　276行からのデータ解説

まず，DMAを呼び出す前に$C3を６回DMAに送る。
これがDMAのRESETになる。

$7D	＝WR0ベースレジスタ（ポートAからポートBに転送）
$00：$30	＝WR0ポートA開始アドレス（3000$_H$番地）
$FF：$BF	＝WR0ブロックレングス（BFFF$_H$バイト）
$14	＝WR1ベースレジスタ（ポートA＝メモリ，インクリメント，可変サイクル未使用）
$18	＝WR2ベースレジスタ（ポートB＝I/O，インクリメント，可変サイクル未使用）
$AD	＝WR4ベースレジスタ（コンティニュアスモード）
$00：$40	＝WR4ポートB開始アドレス（4000$_H$番地）
$9A	＝WR5ベースレジスタ（エンドオブブロックで終結）
$CF	＝ロード（位置について）
$B3	＝フォースレディ（よ〜い）
$87	＝イネーブルDMA（ドン！）



光：このプログラムではメモリを8Kバイトごとに登録するんですけど，最初に一括してグラフィックRAMに送り込んじゃうんですよ。

Yo：それで？

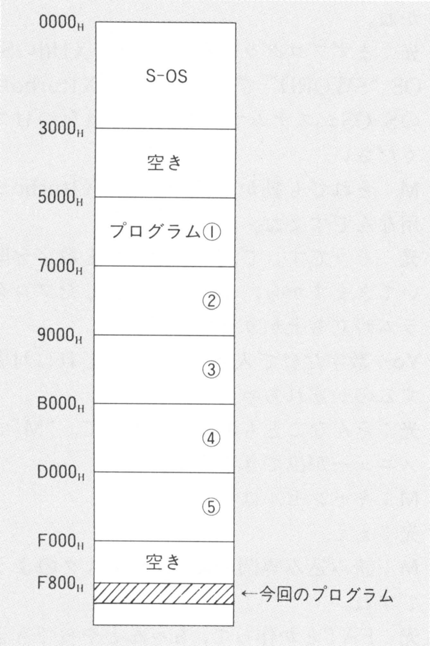
光：5000_{H}番地にひとつ目のプログラム，7000_{H}番地に２つ目，以下9000_{H}番地，$B000_{H}$番地，$D000_{H}$番地にそれぞれプログラムを置いておくんですよ。

Yo：どーすりゃええんじゃ〜っ！

M：芸風変わったね，ようこちゃん。

光：IPLから立ち上がる専用のシステムにしちゃうのがいちばんでしょう。図１のようにプログラムを配置して，まとめてセーブしちゃえばいいんですよ。

図1　メモリマップ



余った8Kバイトの行方

M：そういえば8Kバイトのプログラムが５つで40Kバイトになるよね。あとの8Kバイトはなんに使ってるのかな。

光：するどいですね，マスター。実はツインスクリーンにしてあるんですよ。

Yo：ツインスクリーン？

光：“^P”で画面切り替えができるようになったんですよ。

Yo：画面切り替えって40×25のときにやれるんじゃないの？

光：画面をグラフィックRAMの使ってない部分にセーブしておくんだ。だから80×25のモードでも大丈夫だよ。

Yo：ふうん。

光：こっちもDMAを使っているから高速に切り替わるよ。

M：デバッグのときとかに便利そうだね。

光：ええ，２画面あればいいなあって思ってたんですよ。だからついでに作っちゃったってわけです。

Yo：“^Q”と“^P”を２つ組み合わせて使うとなかなか便利になるのね。

老：オナゴはこうでなきゃいかん！

Yo：長老ってまだ話してたのね。

老：おぬしら，ワシの話をちゃんと聞いておったか？

Yo：ええ，熱心にね。

光：それじゃあ今月はこれくらいで。

Yo：そうね。わたしもクリスマスに備えてこれでプログラミングしてみるわ。

―つづく―

リスト

```
0000                ;       turbo KIT   for X1 turbo
0000                ;
0000                ;       by Hikaru Minamoto
0000
F800                        ORG     $F800
F800
F800                ;
F800                ;  Transfer Program
F800                ;
F800
F800                CTRLP   EQU     $0089
F800                CTRLQ   EQU     $008B
F800                MEMAX   EQU     $1F6A
F800                COLD    EQU     $0000
F800
F800                TRNS
F800 F3                     DI
F801 01 FF 1F               LD      BC,$2000-1
F804 11 01 30               LD      DE,$3001
F807 21 00 30               LD      HL,$3000
F80A 36 20                  LD      (HL),$20
F80C ED B0                  LDIR
F80E CD 53 F8               CALL    BANK1
F811                        ;
F811 CD FF F9               CALL    DMA_RESET
F814 21 16 FA               LD      HL,DATA1
F817 CD 0C FA               CALL    SET_DMA
F81A                        ;
F81A CD 4B F8               CALL    BANK0
F81D 21 FF F7               LD      HL,$F7FF
F820 22 6A 1F               LD      (MEMAX),HL
F823 21 33 F8               LD      HL,CBANK
F826 22 89 00               LD      (CTRLP),HL
F829 21 5B F8               LD      HL,ARGO
F82C 22 8B 00               LD      (CTRLQ),HL
F82F FB                     EI
F830 C3 00 00               JP      COLD
F833
F833                ;
F833                ;  Twin Screen Program
F833                ;
F833
F833                CBANK
F833 CD 53 F8               CALL    BANK1
F836 CD FF F9               CALL    DMA_RESET
F839                        ;
F839 21 24 FA               LD      HL,DATA2
F83C CD 0C FA               CALL    SET_DMA
F83F                        ;
F83F 21 32 FA               LD      HL,DATA3
F842 CD 0C FA               CALL    SET_DMA
F845                        ;
F845 21 40 FA               LD      HL,DATA4
F848 CD 0C FA               CALL    SET_DMA
F84B                BANK0
F84B 01 D0 1F               LD      BC,$1FD0
F84E 3E 60                  LD      A,$60    ; HI-RESO = $61
F850 ED 79                  OUT     (C),A
F852 C9                     RET
F853                BANK1
F853 01 D0 1F               LD      BC,$1FD0
F856 3E 70                  LD      A,$70    ; HI-RESO = $71
F858 ED 79                  OUT     (C),A
F85A C9                     RET
F85B
F85B                ;
F85B                ;       ARGO Key Program
F85B                ;
F85B
F85B                #KBFAD  EQU     $1F76
F85B                #MPRINT EQU     $1FE2
F85B                #LTNL   EQU     $1FEE
F85B                #PRINT  EQU     $1FF4
F85B                #MSX    EQU     $1FE5
F85B                #PRTHL  EQU     $1FBE
F85B                #HLHEX  EQU     $1FB2
F85B                #GETL   EQU     $1FD3
F85B                #CSR    EQU     $2018
F85B                #LOC    EQU     $201E
F85B                #FLGET  EQU     $2021
F85B
F85B                ARGO
F85B CD E2 1F               CALL    #MPRINT
F85E 57 68 69               DM "Which number 1-5  (cansel 0)"
F861 63 68 20
F864 6E 75 6D
F867 62 65 72
F86A 20 31 2D
F86D 35 20 20
F870 28 63 61
F873 6E 73 65
F876 6C 20 30
F879 29
F87A 0D 00                  DB $0D:$00
F87C                ARGO1
F87C CD 21 20               CALL    #FLGET
F87F FE 4D CA               IF    A='M' JP FILES
F882 B2 F9
F884 FE 45 CA               IF    A='E' JP MAKE
F887 C6 F8
F889 FE 30 C8               IF    A='0' RET
F88C FE 36 D2               IF    A>=$36 JP ARGO1
F88F 7C F8
F891 FE 30 DA               IF    A<$30  JP ARGO1
F894 7C F8
F896 D6 30                  SUB     $30
F898 47                     LD      B,A
F899 C5                     PUSH    BC
F89A 21 50 FA               LD      HL,STADR
F89D 3E 40                  LD      A,$40
F89F 0E 20                  LD      C,$20
F8A1                ARGO2
F8A1 81                     ADD     A,C
F8A2 10 FD                  DJNZ    ARGO2
F8A4 77                     LD      (HL),A
F8A5 C1                     POP     BC
F8A6 CD 57 F9               CALL    FILADR
F8A9 11 0E 00               LD      DE,14
F8AC 19                     ADD     HL,DE
F8AD 56                     LD      D,(HL)
F8AE 23                     INC     HL
F8AF 5E                     LD      E,(HL)
F8B0 EB                     EX      DE,HL
F8B1 22 56 FA               LD      (STADR2),HL
F8B4 E5                     PUSH    HL
F8B5 CD 53 F8               CALL    BANK1
F8B8                        ;
F8B8 CD FF F9               CALL    DMA_RESET
F8BB 21 4E FA               LD      HL,DATA5
F8BE CD 0C FA               CALL    SET_DMA
F8C1                        ;
F8C1 CD 4B F8               CALL    BANK0
F8C4 E1                     POP     HL
F8C5 E9                     JP      (HL)
F8C6
F8C6                MAKE
F8C6 CD E2 1F               CALL    #MPRINT
F8C9 4D 65 6E               DM      "Menu input for ARGO KEY" DB $0D
```

```
F8CC 75 20 69
F8CF 6E 70 75
F8D2 74 20 66
F8D5 6F 72 20
F8D8 41 52 47
F8DB 4F 20 4B
F8DE 45 59 0D
F8E1 45 64 69             DM      "Edit number " DB $00
F8E4 74 20 6E
F8E7 75 6D 62
F8EA 65 72 20
F8ED 00
F8EE            MAIN
F8EE CD 21 20             CALL    #FLGET
F8F1 FE 30                CP      '0'
F8F3 CA EE 1F             JP      Z,#LTNL
F8F6 DA EE 1F             JP      C,#LTNL
F8F9 FE 36                CP      '6'
F8FB 30 F1                JR      NC,MAIN
F8FD CD F4 1F             CALL    #PRINT
F900 D6 30                SUB     30H
F902 CD EE 1F             CALL    #LTNL
F905 CD EE 1F             CALL    #LTNL
F908 47                   LD      B,A
F909 CD 57 F9             CALL    FILADR
F90C CD E2 1F             CALL    #MPRINT
F90F 46 69 6C             DM      "File name "
F912 65 20 6E
F915 61 6D 65
F918 20
F919 00                   DB      0
F91A CD 27 F9             CALL    FILDSP
F91D CD 61 F9             CALL    FILIN
F920 23                   INC     HL
F921 CD 8A F9             CALL    ADRIN
F924 C3 C6 F8             JP      MAKE
F927            FILDSP
F927 E5                   PUSH    HL
F928 CD 51 F9             CALL-   FILPRT
F92B 62                   LD      H,D
F92C 6B                   LD      L,E
F92D 11 0E 00             LD      DE,0014
F930 19                   ADD     HL,DE
F931 CD 36 F9             CALL    ADRPRT
F934 E1                   POP     HL
F935 C9                   RET
F936            ADRPRT
F936 CD E2 1F             CALL    #MPRINT
F939 20 3A 20             DM      " : Start ADRS $" DB 0
F93C 53 74 61
F93F 72 74 20
F942 41 44 52
F945 53 20 24
F948 00
F949 7E                   LD      A,(HL)
F94A 23                   INC     HL
F94B 6E                   LD      L,(HL)
F94C 67                   LD      H,A
F94D CD BE 1F             CALL    #PRTHL
F950 C9                   RET
F951            FILPRT
F951 54                   LD      D,H
F952 5D                   LD      E,L
F953 CD E5 1F             CALL    #MSX
F956 C9                   RET
F957            FILADR
F957 21 4C FA             LD      HL,FILENM-16
F95A 11 10 00             LD      DE,16
F95D            FILNM1
F95D 19                   ADD     HL,DE
F95E 10 FD                DJNZ    FILNM1
F960 C9                   RET
F961            FILIN
F961 E5                   PUSH    HL
F962 CD 18 20             CALL    #CSR
F965 2E 0A                LD      L,0AH
F967 CD 1E 20             CALL    #LOC
F96A ED 5B 76             LD      DE,(#KBFAD)
F96D 1F
F96E CD D3 1F             CALL    #GETL
F971 38 3B                JR      C,OUTIN
F973 EB                   EX      DE,HL
F974 16 00                LD      D,0
F976 19                   ADD     HL,DE
F977 EB                   EX      DE,HL
F978 E1                   POP     HL
F979 06 0D                LD      B,13
F97B            FILIN1
F97B 1A                   LD      A,(DE)
F97C FE 00                CP      0
F97E 20 02                JR      NZ,FILIN2
F980 3E 20                LD      A,20H
F982            FILIN2
F982 77                   LD      (HL),A
F983 13                   INC     DE
F984 23                   INC     HL
F985 10 F4                DJNZ    FILIN1
F987 36 00                LD      (HL),0
F989 C9                   RET
F98A            ADRIN
F98A E5                   PUSH    HL
F98B            ADRIN1
F98B CD 18 20             CALL    #CSR
F98E 25                   DEC     H
F98F 2E 26                LD      L,38
F991 CD 1E 20             CALL    #LOC
F994 ED 5B 76             LD      DE,(#KBFAD)
F997 1F
F998 CD D3 1F             CALL    #GETL
F99B 38 11                JR      C,OUTIN
F99D EB                   EX      DE,HL
F99E 16 00                LD      D,0
F9A0 19                   ADD     HL,DE
F9A1 EB                   EX      DE,HL
F9A2 CD B2 1F             CALL    #HLHEX
F9A5 38 E4                JR      C,ADRIN1
F9A7 D1                   POP     DE
F9A8 7C                   LD      A,H
F9A9 12                   LD      (DE),A
F9AA 13                   INC     DE
F9AB 7D                   LD      A,L
F9AC 12                   LD      (DE),A
F9AD C9                   RET
F9AE            OUTIN
F9AE E1                   POP     HL
F9AF C3 C6 F8             JP      MAKE
F9B2            FILES
F9B2 CD E2 1F             CALL    #MPRINT
F9B5 2D 2D 2D             DM      "-------------- MENU --------------"
F9B8 2D 2D 2D
F9BB 2D 2D 2D
F9BE 2D 2D 2D
F9C1 2D 2D 20
F9C4 4D 45 4E
F9C7 55 20 2D
F9CA 2D 2D 2D
F9CD 2D 2D 2D
F9D0 2D 2D 2D
F9D3 2D 2D 2D
F9D6 2D 2D
F9D8 0D 00                DW      000DH
F9DA 3E 01                LD      A,1
F9DC            FILES1
F9DC F5                   PUSH    AF
F9DD 47                   LD      B,A
F9DE CD F3 F9             CALL    NUMBER
F9E1 CD 57 F9             CALL    FILADR
F9E4 CD 27 F9             CALL    FILDSP
F9E7 CD EE 1F             CALL    #LTNL
F9EA F1                   POP     AF
F9EB 3C                   INC     A
F9EC FE 06                CP      6
F9EE 38 EC                JR      C,FILES1
F9F0 C3 5B F8             JP      ARGO
F9F3            NUMBER
F9F3 C6 30                ADD     A,30H
F9F5 CD F4 1F             CALL    #PRINT
F9F8 CD E2 1F             CALL    #MPRINT
F9FB 2E 20 00             DM      ". " DB 0
F9FE C9                   RET
F9FF
F9FF            ;         SUB ROUTINE
F9FF
F9FF            DMA_RESET
F9FF 01 80 1F             LD      BC,$1F80
FA02 3E C3                LD      A,$C3
FA04 16 06                LD      D,6
FA06 ED 79                OUT     (C),A
FA08 15                   DEC     D
FA09 20 FB                JR      NZ,DMA_RESET+7
FA0B C9                   RET
FA0C            SET_DMA
FA0C 16 0E                LD      D,$0E
FA0E 7E                   LD      A,(HL)
FA0F ED 79                OUT     (C),A
FA11 23                   INC     HL
FA12 15                   DEC     D
FA13 20 F9                JR      NZ,SET_DMA+2
FA15 C9                   RET
FA16
FA16            ;         DMA DATA
FA16
FA16            DATA1
FA16 7D 00 30             DB $7D:$00:$30:$FF:$BF
FA19 FF BF
FA1B 14                   DB $14
FA1C 18                   DB $18
FA1D AD 00 40             DB $AD:$00:$40
FA20 9A                   DB $9A
FA21 CF B3 87             DB $CF:$B3:$87
FA24            DATA2
FA24 7D 00 30             DB $7D:$00:$30:$CF:$07
FA27 CF 07
FA29 1C                   DB $1C
FA2A 18                   DB $18
FA2B AD 00 50             DB $AD:$00:$50
FA2E 9A                   DB $9A
FA2F CF B3 87             DB $CF:$B3:$87
FA32            DATA3
FA32 7D 00 40             DB $7D:$00:$40:$CF:$07
FA35 CF 07
FA37 1C                   DB $1C
FA38 18                   DB $18
FA39 AD 00 30             DB $AD:$00:$30
FA3C 9A                   DB $9A
FA3D CF B3 87             DB $CF:$B3:$87
FA40            DATA4
FA40 7D 00 50             DB $7D:$00:$50:$CF:$07
FA43 CF 07
FA45 1C                   DB $1C
FA46 18                   DB $18
FA47 AD 00 40             DB $AD:$00:$40
FA4A 9A                   DB $9A
FA4B CF B3 87             DB $CF:$B3:$87
FA4E            DATA5
FA4E 7D 00                DB $7D:$00
FA50            STADR
FA50 40 FF 1F             DB $40:$FF:$1F
FA53 1C                   DB $1C
FA54 10                   DB $10
FA55 AD                   DB $AD
FA56            STADR2
FA56 00 30 9A             DB $00:$30:$9A
FA59 CF B3 87             DB $CF:$B3:$87
FA5C
FA5C            FILENM
FA5C 57 49 4E             DM      "WINNER S.K    "  DB 0 DW 0030H
FA5F 4E 45 52
FA62 20 53 A4
FA65 4B 20 20
FA68 20 00 30
FA6B 00
FA6C 5A 45 44             DM      "ZEDA III      "  DB 0 DW 0030H
FA6F 41 20 49
FA72 49 49 20
FA75 20 20 20
FA78 20 00 30
FA7B 00
FA7C 42 50 20             DM      "BP 2          "  DB 0 DW 0050H
FA7F 32 20 20
FA82 20 20 20
FA85 20 20 20
FA88 20 00 50
FA8B 00
FA8C 44 52 45             DM      "DREAM         "  DB 0 DW 0030H
FA8F 41 4D 20
FA92 20 20 20
FA95 20 20 20
FA98 20 00 30
FA9B 00
FA9C 4D 41 43             DM      "MACINTO-C B0-"  DB 0 DW 00B0H
FA9F 49 4E 54
FAA2 4F 2D 43
FAA5 20 42 30
FAA8 2D 00 B0
FAAB 00
OBJECT CODE END FAAB
```

Oh!X INDEX'92

特集

SX-WINDOWの未来 ……………………………… 1, 89
SX-WINDOW高解像度の世界 ……………………… 1, 27
どうしてウィンドウなのか？
ウィンドウ環境のために………………………… 1, 90
なにが必要なのか？
SX-WINDOWの可能性 ………………………… 1, 92
X68000の場合＝これまでとこれからを考える
SX-WINDOWを検証する ………………………… 1, 94
オペレーションとデータの統一
Macintosh OSに学ぶ ……………………………… 1, 96
NeXTの場合＝ユーザーインタフェイスを極める
SX-WINDOWはNeXTの夢を見れるか？ ………… 1, 100
AMIGAの場合＝破天荒なマシンには破天荒なウィンドウシステム
飛びそうに軽い，ワークベンチ ………………… 1, 102
GUIを生かすために
グラフィック資産の管理と活用法……………… 1, 106
ウィンドウでのプログラム
これからのプログラミング環境………………… 1, 110
ウィンドウプログラムへの招待
SXアプリケーションの基本構造 ……………… 1, 115
プログラミングスタイルは変わるのか？
ウィンドウとオブジェクト指向………………… 1, 122
2Dグラフィックの拡張 ……………………………… 2, 29
2Dグラフィック処理入門
映像は加工される………………………………… 2, 33
Z's-EXの拡張（その1）
発表Z's-EX ver.1.1 ……………………………… 2, 35
Z's-EXの拡張（その2）
外部ファイルの構成と拡張……………………… 2, 41
基礎からの画像処理
平滑化フィルタの作成…………………………… 2, 46
Z's-EX ver.1.1の外部コマンド作成法
拡張用スケルトンを作る………………………… 2, 52
SCSIの活用 ……………………………………… 3, 81
インタフェイスの基礎知識
SCSIとはなにか…………………………………… 3, 82
大容量を目指せ
ハードディスク購入の心得……………………… 3, 88
とりあえずつなぐ
CD-ROMと光磁気ディスク ……………………… 3, 93
リムーバブルハードディスク
合言葉はSyQuest ………………………………… 3, 96
ディスク共有の試み
SCSIによるX68000直結実験 …………………… 3, 100
成熟するゲームと日本の文化……………………… 4, 89
AIIIの箱庭感覚と日本経済
濡れ手で粟のバブル経済………………………… 4, 90
RPG－箱庭に続くゲームは
感情移入の快感と次世代のゲーム……………… 4, 94
Outside the Game
いまどきのゲームシステムを探る……………… 4, 98
システムソフトと光栄の戦略
シミュレーションは誰のもの？………………… 4, 103
アーケードの移植にもお国柄の違い
異人さんの移植は異色…………………………… 4, 106
スポーツゲームに見るこだわり方の差異
リアルなルールか，SD選手か？ ……………… 4, 108
脱ぎ麻雀とストリップポーカー
快楽のマテリアル………………………………… 4, 110
B級NINJAムービーとインチキNIPPONゲーム
頭にピストルを乗せた人々……………………… 4, 113
どちらも立派な文化のひとつだ
漫画とゲームの微妙な関係……………………… 4, 116
明日のための環境づくり………………………… 5, 51
まずは概念を捉えよう
コンピュータと人間のミゾを埋めるもの……… 5, 52
これがないと始まらない
最低限の道標コマンド…………………………… 5, 55
ハードウェア構成別環境対策
ナベには必ずフタがある………………………… 5, 59
自分だけの環境をつくるために
CONFIG.SYSとAUTOEXEC.BATのからくり …… 5, 62
超空間美術論……………………………………… 7, 60
実録TORNADO秘話
4DCGへの招待 …………………………………… 7, 64
積み木感覚のCGツール
3次元おもちゃ箱「TOYBOX」 ………………… 7, 71
MIRAGE System　Model Stuff
レイトレースの新体系を見る…………………… 7, 75
2次元空間の3次元処理
パースペクティブの高速化……………………… 7, 77
プログラミング再入門…………………………… 8, 81
まず本質を知る
プログラミング言語の前に……………………… 8, 82
フローチャートによるアイデアのまとめ
プログラミングの流れをつかもう……………… 8, 84
BASICで作るDōGAフレームファイル
正しい花瓶の落とし方…………………………… 8, 90
オブジェクト指向に学ぶ
作り散らかせます………………………………… 8, 98
比較的大きなプログラムの独断的制作法
ちょっと大きいモノを書こう…………………… 8, 105
数値演算の熱い逆襲……………………………… 9, 73
カラー紹介　数値演算高速化の世界…………… 9, 72
微積分をシミュレートする
夏休みの最小2乗法……………………………… 9, 74
モデル化による演算高速化
疑似メタボールで遊ぶ…………………………… 9, 76
V70ボードの活用
AFPPを使った3Dグラフィック ……………… 9, 82
68881の性能を引き出す
FPP.MACの作成 ………………………………… 9, 90
68881＋68881＝137762？
68881の並列駆動への挑戦 ……………………… 9, 99
DTMへの招待 …………………………………… 10, 75
概論
MIDIをめぐる環境'92 …………………………… 10, 76
拡張された内蔵音源を使う
Z-MUSICとPCM8 ……………………………… 10, 78
Z-MUSICによる
ローランド音源の基礎知識……………………… 10, 80
市販アプリケーションを使う
MUSIC PRO-68K［MIDI］と仲よくなろう …… 10, 83
Z-MUSIC用MMLデータプリプロセッサ
ZPP.X ……………………………………………… 10, 86
おまけ
Z-MUSICでバビンチョ（ゲーム応用編） ……… 10, 98
ゲームマネージメント…………………………… 11, 73
概論
ゲームシステムの構成…………………………… 11, 74
よりよいゲームを作るために
違いのわかるアソビロジー……………………… 11, 76
リアルタイムゲームのための
"基本"スプライトルーチンの作り方………… 11, 78
ゲームマネージャの作成
BEMSによるキャラクタ管理 …………………… 11, 85
考え方と実例を探って
オブジェクト指向を取り入れる………………… 11, 93
シナリオ管理の方法
アドベンチャーシステムを考える……………… 11, 97
プログラム速度管理関数
XVI.FNC …………………………………………… 11, 101
BASICのスプライト処理高速化
SPRITE2.FNC ……………………………………… 11, 103
グラフィック画面の3D回転
3DRT_256 ………………………………………… 11, 107

特別企画

バーコードの秘密を探る
バーコードバトラーの解析………………………… 4, 46
第7回言わせてくれなくちゃだワ
カラーイラスト大集合
Oh!X reader'sぎゃらりぃ ……………………… 5, 18
micro Communication
言わせてくれなくちゃだワ……………………… 5, 89
1991年度Oh!Xイラスト大賞 …………………… 5, 105
どんな悩みもすっきり解決　質問箱スペシャル… 5, 106
創刊10周年特別企画「Oh!MZ, Oh!X10年間の歩み」 … 6, 97
OLD TIMES Oh!MZ, Oh!X 1982-1991 ………… 6, 98
特別寄稿"なんか言わせてくれなくちゃだワ" …… 6, 108
SENTINELとともに ……………………………… 6, 114
愛読者特大プレゼント…………………………… 6, 38
続・創立10周年特別企画「Oh!MZ, Oh!X10年間の歩み」
連載のすべて（前編　1982－1985）……………… 7, 94
GAME OF THE DECADE ………………………… 7, 98
特別寄稿"なんか言わせてくれなくちゃだワ" …… 7, 102
10年分のSTUDIO MZ・X ……………………… 7, 106
続々・創立10周年特別企画「Oh!MZ, Oh!X10年間の歩み」
連載のすべて（後編　1986－1991）……………… 8, 63
Oh!X 5周年特別企画
愛読者特別モニタ大募集………………………… 12, 49
ショートプロ大集合……………………………… 12, 73
見つけりゃ得点，外せば減点
Where is Mistake? ……………………………… 12, 74
SV.X支援ツール
IFF, ILBMのファイル情報を読む …………… 12, 76
ファイルから文字列を探し出す
STRFIND.C ……………………………………… 12, 78
タイムトライアル3Dゲーム
MAGICAL TRIANGLE ………………………… 12, 80
乱数を使った
地形表示プログラム…………………………… 12, 82
最近流行の不思議議図形
裸眼立体視（ランダムドット風）……………… 12, 84
対戦型シューティング
撃ち合いゲーム………………………………… 12, 86
任意の点を滑らかにつなぐ
高次方程式のグラフを描く…………………… 12, 88
電源システムの管理
パワーダウンマネージャ……………………… 12, 90
電卓インベーダーゲームを再現する
あなたと私の電卓物語………………………… 12, 93
Oh!XとOh!Xの読者の統計 ……………………… 12, 96

特別付録ディスク

創刊10周年記念PRO-68K ………………………… 6, 41
カラー紹介　創刊10周年記念PRO-68K ………… 6, 12
書体倶楽部用フォントエディタ　ZsFNT.X ……… 6, 44
3Dシューティングゲーム　SION II ……………… 6, 49
マルチチャンネルAD PCMドライバ　PCM8 …… 6, 52
アニメーションビュア　SV.X …………………… 6, 57
至高のスプライトメーカー　SM.X ……………… 6, 63
カラー紹介　特別付録　創刊10周年記念PRO-68K …… 6, 12
続・創刊10周年記念PRO-68K
MAGICの料理法　MAGICとSION IIの関係 ……… 7, 52
続々・創刊10周年記念PRO-68K

Z-MUSICシステムver.1.10 ……………………………8, 49
SIONIIを読む人に贈る ゲーム内部のイロハ ………8, 56
超高速ライフゲーム LIFE110.X…………………………8, 60
特別付録
DōGA CGAシステムver.2.50&お試しディスク ……7, 30
発表! DōGA CGAシステムver.2.50……………………7, 32

THE SOFTOUCH

THE SOFTOUCH SPECIAL
1991年度GAME OF THE YEAR ノミネート作品発表 …2, 22
決定! 1991年度GAME OF THE YEAR……………………4, 27
やっぱり勝手にGAME OF THE YEAR……………………4, 33
我々が選ぶベストゲーム……………………………………4, 35
SOFTWARE INFORMATION
ファーストクイーンII/レミングス/スピンディジーII/ジェノサイド2/ヘビーノヴァ/ブルトン・レイ シナリオ集 VOL.3/ユニオン ……………………………………………………1, 28
グラディウスII/シムアース/大戦略III'90/ファーストクイーンII/スタートレーダー/スピンディジーII/ロードス島戦記福神漬……………………………………………………2, 81
マスターオブモンスターズII/F29/F15ストライクイーグルIIシナリオ/スーパー上海ドラゴンズアイ/麻雀遊園地 …………3, 30
ピーター・モリニュー/ポピュラスIIの概要3,32 ……………3, 32
エイリアンシンドローム/苦胃頭捕物帳/ヘビーノヴァ/超人/ドラゴンナイトIII ………………………………………………4, 73
ファイナルファイト/ライフ&デス/ロイヤルブラッド/将棋聖天/ドラゴンストライク ……………………………………………5, 24
ナムコ・ワンダーエッグ/コナミ インタビュー………………5, 26
エトワールプリンセス/棋太平68K/ヴェルスナーグ戦乱/シムアース/ファイナルファイト………………………………………6, 22
ポピュラスII/バトルテック/シュートレンジ/ファイナルファイト ………………………………………………………………7, 18
ライジングサン/バトル/沈黙の艦隊/ふしぎの海のナディア/チェイスH.Q ………………………………………………………8, 22
デスブレイド/リーディングカンパニー/サークII/チェイスH.Q/SX-WINDOW開発キット ………………………………………9, 20
エトワールプリンセス/キャッスルズ/ヴェルスナーグ戦乱/シュートレンジ/CHART PRO-68K ……………………………10, 18
ストライダー飛竜/ロードス島戦記II/バーンウェルト/極/エトワールプリンセス ………………………………………………11, 14
エトワールプリンセス/ストライダー飛竜/究極タイガー/パチンコワールド/ストライクレンジ/幻影都市 ………………………12, 24
ウワサのソフトウェア
POPULOUS II/TEAM SUZUKI …………………………2, 84
メガロマニア/Human Design ……………………………3, 34
Formula One Grand Prix/Senery Animator ……………5, 28
EPIC/OUT OF THIS WORLD ………………………………8, 24
Virtual Reality Studio/GUY SPY …………………………9, 22
AQUAVENTURA/RACE DRIVIN' ……………………………10, 20
ASHES OF EMPIRE/4D SPORTS DRIVING …………………11, 16
Shadow of the BeastIII/D/GENERATION…………………12, 26
GAME REVIEW
出たな!!ツインビー……………………………………………1, 30
ブリッツクリーク………………………………………………1, 34
飛翔鮫……………………………………………………………1, 36
NIKO² ……………………………………………………………1, 38
ラストバタリオン………………………………………………1, 39
PITAPAT …………………………………………………………1, 40
ディノランド……………………………………………………1, 41
ジェノサイド2…………………………………………………2, 86
ワールドゴルフIII………………………………………………2, 89
アルシャーク……………………………………………………2, 90
コード・ゼロ……………………………………………………2, 92
XENON2 …………………………………………………………2, 94
ユニオン…………………………………………………………2, 96
ブルトン・レイ シナリオ集 VOL.3 …………………………2, 97
グラディウスII…………………………………………………3, 36
レミングス………………………………………………………3, 40
大戦略III'90 ……………………………………………………3, 44
伊忍道……………………………………………………………3, 46
ファーストクイーンII…………………………………………4, 76
マスターオブモンスターズII…………………………………4, 78
スーパー上海ドラゴンズアイ…………………………………4, 80
スタートレーダー………………………………………………4, 82
F15ストライクイーグルII シナリオ…………………………4, 83
グラディウスII…………………………………………………4, 84
エイリアンシンドローム………………………………………5, 30
苦胃頭捕物帳……………………………………………………5, 32
超人………………………………………………………………5, 34
ヘビーノヴァ……………………………………………………5, 35
レミングス………………………………………………………5, 36
スピンディジーII………………………………………………6, 26
ロイヤルブラッド………………………………………………6, 28
ノア………………………………………………………………6, 30
ドラゴンストライク……………………………………………6, 32
ライフ&デス……………………………………………………6, 34
将棋聖天&棋太平68K…………………………………………7, 21
シムアース………………………………………………………7, 24
太閤立志伝………………………………………………………7, 26
ウルティマVI……………………………………………………8, 26
三國志III…………………………………………………………8, 28
バトルテック……………………………………………………8, 30
シムアース………………………………………………………8, 32
ファイナルファイト……………………………………………9, 24
ライジングサン…………………………………………………9, 28
ヨーロッパ戦線…………………………………………………9, 30
シューティング68K GAMES……………………………………9, 32
ポピュラスII……………………………………………………10, 22
リーディングカンパニー………………………………………10, 26
ネクタリス………………………………………………………10, 28
サークII…………………………………………………………10, 30
キャッスルズ……………………………………………………11, 18
シュートレンジ…………………………………………………11, 20
ポピュラスII ……………………………………………………11, 22
サンダーレスキュー……………………………………………11, 25
デスブレイド……………………………………………………12, 28
ふしぎの海のナディア…………………………………………12, 30
ロードス島戦記II………………………………………………12, 32
ムーンクレスタ/テラクレスタ ………………………………12, 34
バーンウェルト…………………………………………………12, 36
AFTER REVIEW
パロディウスだ!………………………………………………1, 42
ワールドスタジアム/生中継68 ………………………………2, 98
A列車で行こうIII………………………………………………3, 48
イース……………………………………………………………4, 86
スターウォーズ…………………………………………………5, 38
出たな!!ツインビー……………………………………………6, 36
ジェノサイド2…………………………………………………7, 28
グラディウスII…………………………………………………8, 34
レミングス………………………………………………………9, 34
SION II …………………………………………………………10, 32
シムアース………………………………………………………11, 26
三國志III…………………………………………………………12, 38

活用・レポート

MT-32リズムパート活用講座
LA音源を活用しよう ……………………………………1, 47
MAGICver.2.0対応ゲーム
3D MAZE …………………………………………………1, 132
Z-MUSIC支援ツール
ZPDセパレータZPDCON.X………………………………3, 102
Z's-EX用拡張コマンド
MASK_reverse.X …………………………………………3, 105
Z's-EX&Z'sSTAFF
共用外部ファイル作成法…………………………………7, 48
Z'sSTAFF&Z's-EX用外部ファイル
ジャギー除去に挑戦………………………………………9, 117
ゲーム画面保存プログラム
SAVESC.SYS ………………………………………………11, 32
もうひとつの情報源
海外雑誌の読み方…………………………………………11, 142
SX-WINDOW対応
追いかけっこゲーム………………………………………12, 60
キャラグラゲームのススメ
シュールな風景……………………………………………12, 102
MZ-2500
MATERIAL GIRL ……………………………………………(LIVE)
おまけ ヴェクザシオン ……………………………………(LIVE)
X1/turbo
Where is the Mistake? ………………………………(特別企画)
SCROLL.BAS for X1 ……………………………………(ショート)
THE ENTERTAINER ………………………………………(LIVE)
君が代 ………………………………………………………(LIVE)
IT'S MAGIC …………………………………………………(LIVE)
Danger Line …………………………………………………(LIVE)
出たな!!ツインビーより 風の贈り物 ………………………(LIVE)
美少女戦士セーラームーン …………………………………(LIVE)
スーパーマリオより 地上のテーマ …………………………(LIVE)
ユーフォリーより オープニングテーマ ……………………(LIVE)
X1turbo
michelle plue for X1turbo ………………………………(ショート)

連載・シリーズ

猫とコンピュータ
第66回 ギャラガ for Xmas ……………………………1, 158
第67回 パーソナルが楽しいとき ………………………2, 148
第68回 暖冬日記・たべかけ編 …………………………3, 148
第69回 サーチャー認定試験 ……………………………5, 156
第70回 猫とカラーコピー ………………………………6, 150
第71回 朗読されるパソコン ……………………………7, 156
第72回 新説・猫×7×0.7 ………………………………8, 158
第73回 じいコード,ばあコード …………………………9, 160
第74回 猫に偏差値 ………………………………………11, 154
第75回 肩コリと本棚 ……………………………………12, 150
知能機械概論
第55回 ノイマンはなぜノイマンマシンを作ったのか?
……………………………………………………………1, 155
第56回 マックがあっちへ行く …………………………2, 150
第57回 人の自然な姿と連帯するコンピュータ ………3, 150
第58回 AI新個人主義を支える思想 ……………………4, 164
第59回 神様になる方法 …………………………………5, 150
第60回 生中継「地球最後の瞬間」 ……………………6, 152
第61回 スーパーウォークマンの逆説 …………………7, 158
第62回 なぜ13分で料理が消えたのか …………………8, 162
第63回 近未来型ワープロ序説 …………………………9, 138
第64回 アルゴリズムとしての進化論 …………………11, 152
第65回 アヴァン・ポップで仮想空間から逃げ出せ …12, 148
X-OVER NIGHT
第19回 窓にねずみ ………………………………………2, 152
第20回 本来の使い方 ……………………………………3, 152
第21回 本来のソフト ……………………………………4, 160
第22回 設計の美しさ ……………………………………5, 152
第23回 バーチャル・リアリティ ………………………6, 137
第24回 ダイエーとリクルート …………………………7, 160
第25回 IKEBUKURO ………………………………………8, 160
第26回 新幹線とコンピュータ …………………………9, 136
第27回 パソコンドリームは終わった …………………10, 136
第28回 バックアップ ……………………………………11, 144
第29回 不良資産 …………………………………………12, 152
響子 in CGわ〜るど/ANOTHER CG WORLD
第8回 年賀状 ……………………………………1, 22/160
第9回 時計仕掛けのハート ……………………2, 20/146
第10回 DataSuit ………………………………3, 24/132
第11回 桜散る ……………………………………4, 18/162
第12回 待ち時間 …………………………………5, 20/154
第13回 お帰りなさい ……………………………6, 16/148
第14回 カオス ……………………………………7, 58/154
第15回 羽 …………………………………………8, 18/156
第16回 仮想生物 …………………………………9, 66/158
第17回 キャラクタ ………………………………10, 66/134
第18回 風の吹いた日 ……………………………11, 60/150
第19回 思索カード ………………………………12, 22/146
大人のためのX68000
第16回 明日に向かって3.5インチ ……………………1, 61
第17回 ワープロもいいけど…… ………………………2, 60
第18回 レイアウトの実践 ………………………………3, 57
第19回 大人のためのQuickTime ………………………5, 48
第20回 第3回Oh!Xアンケート分析大会 ………………6, 117
第21回 データ転送で遊ぶ ………………………………7, 84

第22回 パソコン通信に未来はあるか ……………………8, 148
第23回 Communication SX-68K, そしてFIXER …………9, 46
第24回 メモリ食い食いウラ画面 ……………………10, 45
第25回 いよいよ本題, 立体視……………………11, 28
第26回 ついに発売, MATIER ……………………12, 52

X68000マシン語プログラミング
Chapter_1E$_{II}$ 割り込みの上手な活用法 ……………2, 111
Chapter_1F$_{II}$ 常駐プログラムを作る(前編) …………3, 67
Chapter_20$_{II}$ 常駐プログラムを作る(後編) …………5, 77
Chapter_21$_{II}$ テキスト画面の実力 ……………………6, 86
Chapter_22$_{II}$ スプライトを使いこなす …………………8, 133
Chapter_23$_{II}$ 整数演算のアルゴリズム ………………9, 57
Chapter_24$_{II}$ 大きな数の話 ……………………………10, 141
Chapter_25$_{II}$ 探しもの …………………………………12, 107

Oh!X LIVE in '92より
DRAGON SABERより4面 地葷(X68000) ……………1, 71
すき(X68000) ……………………………………………1, 71
THE ENTERTAINER(X1/turbo) ……………………1, 71
ストリートファイターIIより
リュウのテーマ(X68000) ……………………………2, 105
Tide Over(X68000) ……………………………………2, 105
ギャラクシーフォースよりTRY-Z(X68000) ………3, 134
君が代(X1/turbo) ………………………………………3, 134
あじさいのうた(X68000) ………………………………4, 50
ショパン練習曲作品25-2へ短調(X68000) ……………4, 50
IT'S MAGIC(X1/turbo) …………………………………4, 50
フレンズ(X68000) ………………………………………5, 121
Danger Line(X1/turbo) …………………………………5, 121
TURBO OUTRUNより Shake the Street(X68000) ………6, 68
ネメシス2より Ancient relics(X68000) ………………6, 68
対談! GMコンポーザー 第2回「古代祐三」 …………6, 76
Bye Bye My Love(X68000) ……………………………7, 87
MATERIAL GIRL(MZ-2500) ……………………………7, 87
おまけ ヴェクザシオン(MZ-2500) ……………………7, 87
対談! GMコンポーザー 第3回「GAMADELIC」 ……7, 93
ドラゴンセイバーより 氷穴(X68000) ………………8, 140
ガラガラヘビがやってくる(X68000) …………………8, 140
出たな!!ツインビーより 風の贈り物(X1/turbo) ……8, 140
恋をしようよ Yeah! Yeah!(X68000) …………………9, 141
ゆめいっぱい(X68000) …………………………………9, 141
対談! GMコンポーザー 第4回「S.S.T.BAND」 ………9, 146
美少女戦士セーラームーン(X1/turbo) ………………10, 49
笑顔を探して(X68000) …………………………………10, 50
ドラゴンセイバー・エンディングテーマ(X68000) ……10, 52
Weapon・Shop(X68000) ………………………………10, 54
ローリングサンダー2より
TERROR FOR SALE (X68000) ………………………10, 55
ストリートファイターIIより
BALROGのテーマ(X68000) …………………………11, 134
ARCADIA(X68000) ……………………………………11, 134
スーパーマリオより 地上のテーマ(X1/turbo) ………11, 134
LAST CHRISTMAS ………………………………………12, 125
闇の血族より 次回予告のテーマ(X68000) …………12, 125
ユーフォリーより オープニングテーマ(X1/turbo) …12, 125

DōGA・CGアニメーション講座
<21>山越え, 谷越え, どこまでも(前編) ……………1, 64
<最終回>山越え, 谷越え, どこまでも(後編) …………3, 50

DōGA CGアニメーション講座 ver.2.50
みんな準備はいいか? ……………………………………8, 66
第1回 打倒TORNADOへの第一歩(前編) ……………9, 40
第2回 打倒TORNADOへの第一歩(中編) ……………10, 38
第3回 打倒TORNADOへの第一歩(後編) ……………11, 64
第4回 打倒TORNADOへの第一歩(完結編) …………12, 40

TORNADO 芸術祭グランプリへの道
第1回 TORNADO発進準備完了 ………………………11, 72
第2回 各種ツールを使ったモデリング(1) ……………12, 47

よいこのSX-WINDOW講座
第6回 グラフマンで図形を描く …………………………4, 63
第7回 レクタングルで遊ぶ ………………………………5, 137
第8回 リージョンで遊ぶ …………………………………7, 113
第9回 イメージを極める …………………………………8, 68
第10回 スクロールバーに挑戦 …………………………10, 111
第11回 タスク間通信をためす …………………………11, 119

ショートプロぱーてぃ
その28 ヴィジュアルである ……………………………1, 145
珍客万来.X for SX-WINDOW
SHIBAN.BAS for X68000
その29 時の流れは早すぎて ……………………………2, 135
DTIME.BAS for X68000
make cnf.bas for X68000
その30 プログラム武闘派宣言! …………………………3, 143
FS.X for X68000
SCROLL.BAS for X1
その31 なんてたってぎゃんぶる! ………………………4, 58
KEIBA.BAS for X68000
MUKUMUKU.X for X68000
その32 本気でよい! ……………………………………5, 67
michelle plue for X1turbo
ZCMKR.BAS for X68000
その33 戦うディスクなのだ ……………………………6, 143
DBTL.X for X68000
NEW999.BAS for X68000
その34 よく学び, よく遊べ ………………………………7, 120
MFGEN.X for X68000
ReflexAction.bas for X68000
その35 夏です, 金鳥です, 花火です ……………………8, 152
SOSUU.X for X68000
HANABI.X for X68000
その36 書は心を表すものなり …………………………9, 50
SHODO68K.BAS for X68000
WILD.C for X68000
その37 BATCHKEY.XとSLOT.BAS …………………10, 36
BATCHKEY.X for X68000
SLOT.BAS for X68000
その38 文明の窓・テレビであるのだ …………………11, 146
NTV.C (X68000)
EINSTEIN.BAS for X68000
その39 散らかしOK, 片づけOK ………………………12, 132
ICONMAKE.BAS for X68000
CDS.C for X68000

マシン語カクテルin Z80's Bar
第28回 新年会模擬パーティ ……………………………1, 139
第29回 新年会はまだ続く ………………………………2, 130
第30回 涙のゲーム制作日誌(前編) ……………………3, 73
第31回 震撼の敵機登場(ゲーム制作日誌(中編)) ……4, 151
第32回 誘惑の誘爆(ゲーム制作日誌(後編)) …………5, 114
第33回 怒濤の三角関数 …………………………………6, 130
第34回 お城と流れ星ーその1ー ………………………7, 124
第35回 お城と流れ星ーその2ー ………………………9, 123
第36回 お城と流れ星ーその3ー ………………………10, 153
第37回 ユーティリティがほしい ………………………12, 158

吾輩はX68000である
第9回 完成! 簡単アニメーションツール ………………1, 79
第10回 さらなるスクロール ……………………………2, 73
第11回 入出力で復活を …………………………………3, 114
第12回 発見! パレット設定法 …………………………4, 141
第13回 優先順位の決定 …………………………………5, 109
第14回 渡る世間は半透明 ………………………………6, 138
第15回 飛び出せ! ディスク ……………………………8, 38
第16回 時を刻む …………………………………………9, 54
第17回 マウスで簡易アニメーション …………………10, 137
第18回 極楽た〜ぼマウス ………………………………12, 153

X68000CARDDRV用カードゲーム
サバイバル・ゲーム ………………………………………1, 149
Are You Lucky? …………………………………………2, 78
ピラミッド …………………………………………………3, 107
アコーディオン ……………………………………………3, 112
サーティーン・ダウン ……………………………………11, 115

Creative Computer Music入門
(4)楽曲の基本, 終止形 …………………………………1, 52
(5)非和声音の種類と使用法 ……………………………2, 66
(6)基本的なリズムの刻み方 ……………………………3, 137
(7)メロディが生まれるまで ……………………………4, 145
(8)アンサンブルの成り立ち ……………………………5, 146
(9)メロディどうしの係わり ……………………………6, 78
(10)曲を仕上げる …………………………………………7, 145
(11)効率的な採譜のやり方 ………………………………8, 125
(12)偶成和音と借用和音 …………………………………9, 153
(13)ボーカルのかわりに…… ……………………………10, 157
(14)分厚いハーモニーを作る ……………………………11, 129
(15)弦のアレンジ …………………………………………12, 119

ハードウェア工作入門
<19>ハイテクタンク製作(完成編) ……………………1, 58
<20>赤外線リモコン制御(その1) ……………………2, 100
<21>赤外線リモコン制御(その2) ……………………3, 62
<22>赤外線リモコン制御(その3) ……………………4, 137
<23>赤外線リモコン制御(その4) ……………………5, 133
<24>ハードウェア工作の心得(総集編) ………………6, 82

コンピュータアーキテクチャ編
<25>コンピュータの基礎(導入部) ……………………7, 151
<26>論理演算で加算器を作る …………………………8, 45
<27>デジタル論理回路を学ぶ …………………………9, 148
<28>加算器回路の工作実習 ……………………………10, 60
<29>外部電源の製作 ……………………………………11, 125
<30>レジスタ加算器の設計 ……………………………12, 54

全機種共通システム

THE SENTINEL ……………………………………………1, 125
パズルゲームLINER ………………………………………1, 126
THE SENTINEL ……………………………………………2, 117
シミュレーションゲームPOLANYI ………………………2, 118
THE SENTINEL ……………………………………………3, 121
カードゲームKLONDIKE …………………………………3, 122
THE SENTINEL ……………………………………………4, 119
実践Small-C講座(1) オプティマイザO80 ……………4, 120
THE SENTINEL ……………………………………………5, 127
実践Small-C講座(2) COMMAND.OBJ ………………5, 128
THE SENTINEL ……………………………………………6, 123
実践Small-C講座(3) COMMAND.OBJ2 ………………6, 124
THE SENTINEL ……………………………………………7, 129
実践Small-C講座(4) 関数リファレンス ………………7, 130
THE SENTINEL ……………………………………………8, 111
実践Small-C講座(5) ワイルドカード …………………8, 112
グラフィックライブラリ GRAPH.LIB ……………………8, 116
THE SENTINEL ……………………………………………9, 109
O-EDIT & MODCNV ………………………………………9, 110
THE SENTINEL ……………………………………………10, 121
実践Small-C講座(6) SLENDER HUL …………………10, 122
THE SENTINEL ……………………………………………11, 45
実践Small-C講座(7) EDIT ………………………………11, 46
THE SENTINEL ……………………………………………12, 137
実践Small-C講座(8) MAKE ……………………………12, 138

イベント/ギャラリー

イベント
X68000芸術祭 首都圏大会 ………………………………1, 26
第4回アマチュアCGAコンテストレポート ………………5, 46
第1回X68000芸術祭全国大会&見・体・験フェア ………6, 14
マイクロコンピュータショウ&ビジネスショウ'92 …7, 16
第1回全日本X68000芸術祭全国大会スペシャルレポート
感動の神風は2度吹く!?(EYE VS TORNADE) …………7, 42
データショウ'92 ……………………………………………11, 57
SHOW REPORT エレクトロニクスショウ'92 ……………12, 21

Oh!X Graphic Gallery
DōGA CGアニメーション講座 ……………………………1, 24
DōGA CGアニメーション講座 ……………………………3, 21
第4回アマチュアCGAコンテスト入選作品発表 …………4, 20
DōGA CGアニメーション講座 ……………………………10, 65
DōGA CGアニメーション講座 ……………………………11, 63
DōGA CGアニメーション講座 ……………………………12, 20

Oh!X reader'sぎゃらりぃ
あけましておめでとうのコーナー ………………………3, 22
第7回言わせてくれなくちゃだワ
カラーイラスト大集合 ……………………………………5, 18
暑中見舞だ! ………………………………………………10, 68

THE USER'S WORKS
FX予選/RACING CHAMP/R3 GOLF/DAMON SLAYER3/
☆DUST BOX ………………………………………………3, 26
SHOOT DOWN-70/すらいむらいだあ/SimDungeon ……7, 57
BLUE WINGS ………………………………………………9, 65

SX福袋 ……………………………………………11, 62

製品紹介

ハードウェア
GS規格対応音源モジュールCM-300/500 …………………1, 44
3.5インチFDD搭載X68000 Compact XVI ………………4, 24
新製品紹介 X68000 Compact XVI
ハードウェアの概要とメディアのゆくえ…………………4, 38
これがSX-WINDOW ver.2.0だ ……………………………4, 42
GENERAL MIDI対応音源モジュールKORG 03R/W………5, 40
多機能赤外線リモコン制御システムMIC 68K ……………5, 44
試用レポート ついに姿を見せたV70ボード ……………7, 138
GM対応音源モジュール TG100 ……………………………8, 42
試用レポート X68000用CD-ROMドライブを使う ……10, 69
ソフトウェア
速報 MIRAGE Model Stuff ………………………………2, 28
レイアウトソフトPressConductor PRO-68K
DTPへの道は遠いか？ ……………………………………2, 140
SX-WINDOW ver.2.0 ………………………………………4, 26
スプライトエディタ ぴくせる君ver.1.20 ………………4, 88
速報 Z'sSTAFF PRO-68K ver.3.0 …………………………5, 22
追加レポート SX-WINDOW ver.2.0を検証する
ここまできたSX-WINDOW …………………………………5, 73
SXの課題点は？ …………………………………………5, 75
Z'sSTAFF PRO-68K ver.3.0
輝く絵の具・記憶というキャンバス………………………6, 18
多機能グラフィックツールMATIERを使う(前編) ………8, 20
SX-WINDOW対応音色エディタ SOUND SX-68K …………8, 36
多機能グラフィックツールMATIERを使う(後編) ………9, 36
MIRAGE System Model Stuff(2) …………………………9, 68
概要紹介 SX-WINDOW開発キット ………………………10, 34
グラフチャートソフトCHART PRO-68K …………………11, 58
版下作成支援ソフトY-300A ……………………………12, 59

INFORMATION

ペンギン情報コーナー
ISDN対応画像伝送機DT-100(富士写真フイルム) ……1, 162
超高速モデムMulti Modem V32B
(オール・テクノロジー・グループ)……1, 162
ローコスト高速モデムMD96FS5VII(オムロン) ………1, 162
無停電電源装置BU351X(オムロン) ……………………1, 163
ハンディ液晶プロジェクタHP-40Hi
(富士写真フィルム)……1, 163
車載用キット標準装備の液晶テレビET-562R
(セイコーエプソン)……1, 163
書院シリーズWD-SB30(シャープ) ……………………2, 154
電子システム手帳＆カードPA-T1, PA-7C47
(シャープ)……2, 154
漢字辞書内蔵データバンクウォッチDKW-100
(カシオ計算機)……2, 154
画像処理装置HK-700N(ミノルタ)………………………2, 155
業務用追記型CDレコーダCDR-1(日本マランツ)………2, 155
ワープロ"Pen書院" WV-S200(シャープ) ………………3, 154
書院シリーズWD-SD90(シャープ) ……………………3, 154
ビデオストレージシステムMMD-B50(アイワ) …………3, 154
電子システム手帳用ICカードPA-3C41/5C08/09
(シャープ)……3, 155
3.5インチFD セサミストリートシリーズ(ソニー) …3, 155
防磁加工のフロッピーケースioシリーズ
(LIHIT LAB.)……3, 155
音声認識汎用リモコンMIC PRO-68K
(アンフィニーシステム)……4, 166
ポケットモデムMC24PA5(マイクロコア)………………4, 166
電子手帳用データネットワークシステム
PA-9C80/CE-90CM(シャープ)……4, 166
ワイヤレスプリンタリンクP-WAVE(オムロン) ………4, 167
SC-55/CM-64用データ曲集ソングファイルシリーズ
(サン・ミュージカル・サービス)……4, 167
企業向けハイパー電子システム手帳PA-V1(シャープ) 5, 158
ラップマシン「ラップマンRAP-10」(カシオ計算機) 5, 158
電子システム手帳用ICカード
PA-9C3/60/5C06S/10S/3C44S/47S(シャープ) ……5, 158
名刺読み取り機PV-BR1(シャープ) ……………………5, 159
ヒューマニックメモリユニットHM-100/400/2000
(システムサコム)……6, 154
大容量SRAMカードML-8M-PG/4M-PG(日立マクセル) 6, 154
光カードシステムRW-20/OC-20(キヤノン)……………6, 154
ISDNモデムMT96FS5V(オムロン) ………………………6, 155
A4カラーOHPプリンタCP-4000シリーズ"写嬢"
(日本アビオニクス)……6, 155
X68000用増設5インチFDD CZ-6FD5(シャープ)………7, 162
X68000用CD-ROMドライブ KGU-XCD(計測技研) ……7, 162
BTRONパソコン電房具1B/note
(パーソナルメディア)……7, 162
システム手帳サイズインスタントカメラ
ミスター・ハンディーMF(富士写真フイルム)……7, 163
ハンディ液晶プロジェクタHP-M1
(富士写真フイルム)……7, 163
ファイル転送ユーティリティXIN/XOUT II(電机本舗) 7, 163
X68000用ハードディスクEFX-100B/140B
(エニックス)……8, 164
デジタルピアノKR-650(ローランド) …………………8, 164
電子システム手帳用ICカードPA-5C11S/12S(シャープ)
……………………………………………………………8, 164
SX-WINDOW開発ツールキットCZ-288LWD(シャープ) 8, 165
ファックスモデムEPSON MX-240(セイコーエプソン) 8, 165
小型・軽量化の「写ルンです エコノショット」
(富士写真フイルム)……8, 165
X68000用ハードディスクGF-120/200/240/300/500
(ジェフ)……9, 162
電子システム手帳用ICカードPA-3C45S/48S
(ディ・メーレ/ログ)……9, 162
コンパクト液晶ビジョンXV-P1(シャープ) ……………9, 163
ビジネスパーソナルファクシミリUX-11(シャープ) …9, 163
通信機能付きスクリーンエディタXe ver.2
(エル・クラフト)……9, 163
電子タイトルプリンタNS-TB1(コクヨ) ………………10, 162
特定小電力トランシーバICB-U700/RCB-U33(ソニー) 10, 162
コンパクトコピーマシンU-BIX1112(コニカ) …………10, 162
コードレス留守番電話機CJ-A45(シャープ) …………10, 163
FORTHコンパイラMF68K(バージョンアップ)
(マイクロフォース)……10, 163
ハーフハイト光磁気ディスクドライブJY-7500/750
(シャープ)……11, 156
音声電卓・グラフィック関数電卓CS-2600/EL-9300
(シャープ)……11, 156
ステルス型バーコード(日立マクセル)…………………11, 156
光カードリーダ/ライタLC-304(日本コンラックス) …11, 157
X68000用サブMPUボードPOLYPHON
(ネオコンピュータシステム)……12, 166
TFTカラー液晶ディスプレイLQ14D311/M111
(シャープ)……12, 166
超小型無停電電源装置BU251X(オムロン) ……………12, 167
マルチスキャン・フィルムレコーダ"写嬢"FR-2000
(日本アビオニクス)……12, 167
NTTPCネット主催 第6回CGコンテスト ………………1, 163
第1回Z'sGRACON(ツァイト) …………………………1, 163
インフォメーションファクス(NTT) …………………2, 155
DBサービスをPC-VANへ(三菱総研) …………………2, 155
画像圧縮/伸張用超高速DSPを開発(松下電器産業)…4, 167
画像情報生成処理者試験(画像情報教育振興会) ……5, 159
Joshinデスクトップ・ミュージック・パーティ'92
(上新電機) ……6, 155
ネットワークバトルコミュニケーション
ファイナルリコール(エルスタッフ) ……7, 163
リアルタイムRPG「ルビンスタインの鉱山」
(インターブレイン) ……7, 163
GAME MUSIC FESTIVAL'92 ……………………………8, 165
TVゲーム情報ダイヤル(東京電化) ……………………8, 165
ダイヤルQ²アスキー書籍・データライブラリー
(アスキー) ……9, 163
第5回デスクトップ・ミュージック「力作」コンテスト
(ローランド/ローランドMCクラブ) ……10, 163
Sapporo Multimedia&CG'92
(北海道コンピュータグラフィックス協会) ……10, 163
電脳未来展TRONSHOW'92
(トロン・ショウ実行委員会) ……11, 157
論文コンテスト 多様化・飛躍化するPC環境を考える
(化成バーベイタム) ……11, 157
エンタテイメントスクエア チルコポルト
(コナミ/コナミエンタテイメント) ……11, 157
パソコン通信サービスsuperLinks
(日本テレネット)……12, 167
パソコン通信ネットワークEYE-NET
平成教育委員会(フジミックス)……12, 167
ビジュアル通信対戦ゲームGALAXY MIND
(シンフォニック・プロジェクト)……12, 167
FILES Oh!X 新刊書案内
パソコンゲームの達人……………………………………1, 164
コンピュータソフト………………………………………1, 165
GAME ゲーム げぃむ ……………………………………1, 165
人工現実……………………………………………………2, 156
ソフトウェア基地物語……………………………………2, 157
2001年のコンピュータ……………………………………2, 157
アルーア……………………………………………………3, 156
人とコンピュータのサイコロジ…………………………3, 157
ゆらぎ・カオス・フラクタクル…………………………3, 157
「情報基礎」入門編 ………………………………………4, 168
激烈！日米コンピューター業界戦争……………………4, 169
パソコン［界］遊学見聞録………………………………4, 169
創造する機械………………………………………………5, 161
仕事に活かすべんりパソコン術…………………………5, 161
ソフト業界・こんな会社が危ない………………………5, 161
アラン・ケイ………………………………………………6, 157
脱・電脳生活………………………………………………6, 157
CG夢博物館 ………………………………………………6, 157
笑って覚えるコンピュータ事典…………………………7, 165
どうなるコンピュータ業界………………………………7, 165
こんなもんいかがっすかぁ………………………………7, 165
グリフォンズ・ガーデン…………………………………8, 167
バーチャル・テック・ラボ………………………………8, 167
すみやす荘パソコン通信日記……………………………8, 167
情報文化問題集……………………………………………9, 165
遊園地の現在学……………………………………………9, 165
電脳筒井線 朝のガスパール・セッション PART2 ……9, 165
パソコン用語迷解辞典……………………………………10, 165
人間情報社会の実現に向けて……………………………10, 165
オールザットウルトラ科学まんが①……………………10, 165
青空のリスタート…………………………………………11, 159
別冊宝島EX 現代数学で遊ぶ本 …………………………11, 159
PC-PAGE 28 乱調電脳用語事典 …………………………11, 159
プログラマの妻たち………………………………………12, 169
踊るコンピュータ…………………………………………12, 169
コンピュータ社会と漢字…………………………………12, 169

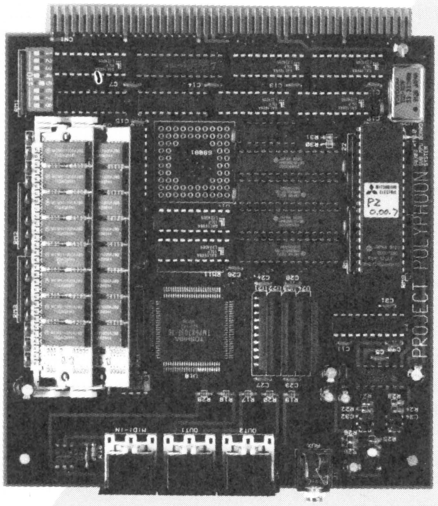
その他
Oh!X INDEX'92 ……………………………………………12, 162
常設コーナー
愛読者プレゼント
ペンギン情報コーナー
FILES Oh!X
Oh!X質問箱
STUDIO X
編集室から/DRIVE ON/ごめんなさいのコーナー/
SHIFT BREAK/microOdyssey
特別付録
5"2HDディスク創刊10周年記念PRO-68K(6月号)
5"2HDディスクDōGA CGAシステムver.2.50＆お試しディスク
(7月号)

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

X68000用サブMPUボード
POLYPHON
ネオコンピュータシステム


POLYPHON

ネオコンピュータシステムでは，X68000用サブボード「POLYPHON」を発売する。

メインMPUとして，16MHzの68000をコアとしたASSP（68HC000を核にDMAC，タイマ/カウンタ，各種コントローラを1チップに収めた特定用途向け標準品）MP68303Fを使用している。

本機には，本体用増設RAM（30ピンSIMMソケット×2），MIDIインタフェイス，コプロセッサ68881/2用ソケットを標準装備。MIDIインタフェイスは，MP68303Fのシリアルインタフェイスを用いたもので，純正MIDIボードとの互換性はない。

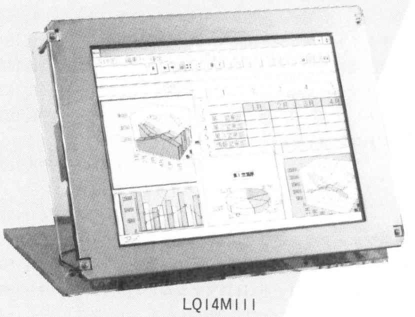
また，MP68303F内蔵のDMACと汎用D/Aの組み合わせによる，分解能8ビット，再生レートDC～100KHz程度のSTEREO PCM機能を搭載。出力はL/R各1チャンネル，左右256段階のパンニングなどはソフト的に行うようになっている。

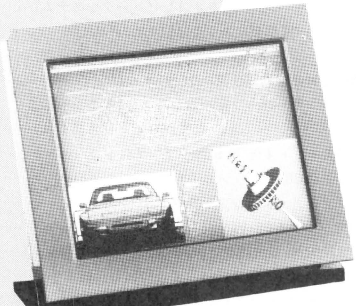
対応アプリケーションとして，「POLYPHON」専用PCM8，RCシステム，Z-MUSICシステム，MLDミュージックドライバなどがあり，本体に同梱されている。

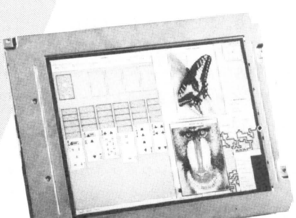
価格は，メインメモリ2Mバイトタイプが65,000円，8Mバイトタイプが90,000円（ともに税別）で，販売は当面の間，直販のみとなっている。

〈問い合わせ先〉
㈱ネオコンピュータシステム　☎03(5682)7007

TFTカラー液晶ディスプレイ
LQ14D311/M111
シャープ


LQ14M111


17型EWS仕様
TFTカラー液晶ディスプレイ


10.4型XGA仕様
TFTカラー液晶ディスプレイ

シャープは，12月より14型OA/AV用デジタル駆動TFTカラー液晶ディスプレイを量産，販売する。

今回，開発，商品化した14型TFTカラー液晶ディスプレイは，6ビット64階調のデジタル多階調ドライバLSIを搭載し，24万色表示対応したOA用ディスプレイ「LQD311」と，デジタル多階調ドライバLSIに入力信号をデジタル画像に高速変換する画像処理インタフェイスを内蔵し，8ビット256階調1670万色表示が可能なAV用ディスプレイ「LQM111」の2機種。

両機種とも640×480ドットの解像度をもち，ドットピッチは0.15mm（横）×0.45mm（縦），モジュール外形サイズが384mm（横）×285mm（縦）×30mm（厚さ），モジュール重量が1,800gである。

サンプル価格は「LQ14D311」が550,000円（税別），「LQ14M111」が620,000円（税別）となっている。

また，同社では17型EWS仕様TFTカラー液晶ディスプレイ，10.4型XGA仕様TFTカラー液晶ディスプレイの開発にも成功した。

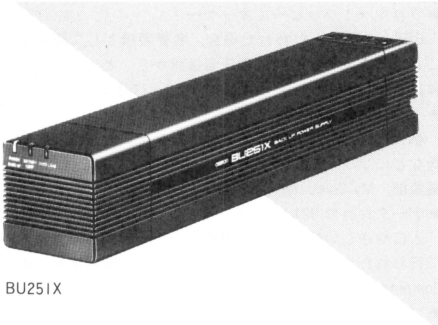
EWS仕様ディスプレイは，1,280×1,024の解像度を実現し，ワークステーション用のモニタとして利用が期待されている。XGA仕様ディスプレイは，現行の10.4型VGA仕様ディスプレイ（同社製「LQ10D021」）と同一サイズで，1,024×768画素をもつディスプレイである。

XGA仕様ディスプレイは，12月からサンプル販売，来年6月より量産販売する予定。

〈問い合わせ先〉
シャープ㈱　☎03(3260)1161,06(621)1221

超小型無停電電源装置
BU251X
オムロン

オムロンでは，電源トラブル発生時にシステムやコンピュータをバックアップする，無停電電源装置「BU251X」を発売した。



BU251X

本機は1990年10月に発売された「BU181X」の後継機種に比べ，出力容量を250VA(165W)にアップしたもので，消費電力165Wのパソコンなら５分間のバックアップができる。

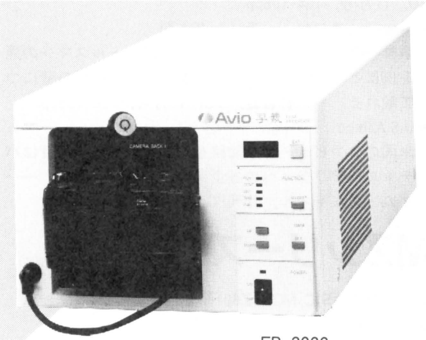
また，2.5時間で80%，５時間で100%の充電が完了する。外形寸法は，76mm(幅)×73mm(高さ)×400mm(奥行)のコンパクトサイズ，重量も3.5kgの軽量設計である。

価格は49,800円（税別）となっている。

〈問い合わせ先〉

オムロン㈱　☎03(5488)3221

マルチスキャン・フィルムレコーダ “写嬢” FR-2000 日本アビオニクス



FR-2000

日本アビオニクスでは，コンピュータの画像イメージを各種フィルムへプリントするフィルムレコーダ「“写嬢” FR-2000」を発売した。

水平走査周波数15～45kHzのディスプレイにマルチスキャンで対応し，1670万色のフルカラーにも対応している。最大解像度は水平1,400ドット，垂直1,050本である。

さらにオートコントラスト，オートブライトネス機能を搭載している。

価格は700,000円(税別)。

〈問い合わせ先〉

日本アビオニクス㈱　☎03(3725)7818

INFORMATION

ビジネスパワーネットワーク super Links 日本テレネット

日本テレネットは，ビジネスパワーネットワーク「super Links」を開始した。

内容は，国内外の100メディアから情報をピックアップする「メディア・ジャック」，ビジネスマンの能力を養うための講座「ネットワーク・ビジネススクール」，自分の適性や能力をチェックする「自己診断シミュレーション」などのサービスを行っている。申し込みは電話，FAX,ハガキで受け付けている。入会金は3,000円，アクセス料は１分10円である。料金の支払いは，すべてJCBかVISAのクレジットカードで行われる。

また，「super Links」開局記念として，抽選で20名の方々に入会金を無料にする，トライアルキャンペーンを実施中である。官製ハガキに住所，氏名，職業，電話番号，現在所有のパソコン機種を明記して，下記の応募先へ郵送すること。締め切りは11月30日必着。

＜応募先＞

〒604　京都府京都市中京区烏丸御池下ルリクルートビル8F　スーパーリンクスネットOh!X係

〈問い合わせ先〉

日本テレネット㈱super Links事務局

☎075(211)2141，FAX075(221)1666

パソコン通信ネットワークEYE-NET 平成教育委員会 フジミックス

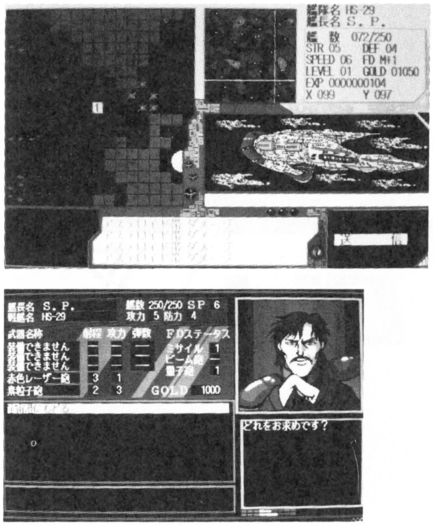
フジミックスはフジサンケイ・パソコン通信ネットワークEYE-NETにおいて，「平成教育委員会」の番組を設けた。

内容は，テレビ版「平成教育委員会」と連動したもので，実際に出題された問題と解答の紹介，そして，EYE-NETの会員からユニークな問題を電子メールで募集している。また，フジテレビ文字放送でも「平成教育委員会」の番組を設け同時進行していく予定。

〈問い合わせ先〉

㈱フジミックス EYE-NET・文字放送センター　☎03(3357)1738

ビジュアル通信対戦ゲーム GALAXY MIND シンフォニック・プロジェクト



写真はPC-9801の画面です

シンフォニック・プロジェクトでは，ダイヤルQ²回線を使ったビジュアル通信対戦ホストを開局した。

ゲームは，GALAXY MINDの世界で繰り広げられる戦術レベルの戦闘をシミュレーションしたもの。基本的にユーザー主体のゲームであり，アクセス状況でゲームが変わっていく。そして，ホスト自身をバージョンアップすることで，端末側のプログラムを変更せずゲーム世界を部分的に変えることもできる。

また，通常のコンピュータ対戦のほかにユーザーどうしが対戦できるSOM（スペースオペラモード）も用意され，毎週金，土曜日には複数のユーザーによる団体戦も行われている。

なお，端末専用ソフトは無料で配布し，ソフト料金をアクセス料金で支払うシステムとなっている。アクセスポイントと同じ市内であれば，１分10円（通話料込み）で遊べ，エンディングまでを最終目標とした場合，通常の人であれば８時間くらいかかるとして，600円(１時間)×８＝4,800円のゲーム料金を支払うことになる。

現在，端末専用ソフトは PC-9801のみ。X68000版の端末専用ソフトは12月に予定されている。

〈問い合わせ先〉

シンフォニック・プロジェクト

☎075(822)3068

FILES Oh!X

このインデックスは，タイトル，注記――著者名，誌名，月号，ページで構成されています。12月はなんといってもクリスマス！　最近はサンタさんも，パソコン使って子供たちとプレゼントのデータベースを作っていたりして……。

一般

▶アルゴリズムを見切ったぞ!?
画面の効果の巻・その１。パソコンソフトには欠かせない「文字表示」を手軽に，リッチに見せるテクニック。――おにおん，テクノポリス，11月号，138-142pp.

▶ど～するど～なる!?　パソコンゲーム！
「美少女ソフトは今……」。ゲームソフトの売り上げ低迷のなか，業界を支えているのがアダルトソフト。摘発事件以降のソフトウェア倫理委員会設立の動きなどを報告。――編集部，テクノポリス，11月号，143-148pp.

▶特集　絶対に知っておきたいコンピューターウイルス基礎知識
迷惑このうえないコンピュータウイルス。「僕だけは大丈夫」と油断しているとひどい目にあうかも。まずは敵を知ろう！　ウイルスの定義や種類，感染ルート，予防法など。――編集部，LOGIN，19号，195-209pp.

▶92　SIGGRAPH
年に一度のCGのお祭り「SIGGRAPH」がシカゴで開催。その出品作のなかから，アート寄りの綺麗な作品を紹介する。――編集部，LOGIN，19号，210-215pp.

▶MUSIC LABO
X68000用MIDIボード「CZ-6BM1」などを紹介。また，アナログシンセサイザの原器で，音作りの仕組みを解説。――編集部，LOGIN，20号，228-235pp.

▶データショウ'92 イベント・レポート
東京・晴海で９月に催されたデータショウ。高速化を目指すパソコン，ペン型入力装置活用への動きなど。――編集部，マイコンBASIC Magazine，11月号，76-77pp.

▶BASICプログラミング講座
中学・高校で学ぶ物理に関係したテーマを選んでプログラミングを勉強。今回は，３次元CGの基礎であるワイヤーフレームの形状をグラフィックを使って表示させるプログラム。３次元変換の学習のヒントになる。――東幸太，マイコンBASIC Magazine，11月号，83-87pp.

▶ワープロ/パソコン通信新聞
オンラインゲーム「Habitat」端末ソフトのバージョンアップ。NIFTY-Serveの「エイズ情報サービス」など大手ネットの近況。市販通信ソフトの紹介など。――山本まさこ，マイコンBASIC Magazine，11月号，94-98pp.

▶新時代をむかえた日本語ワープロ
ソフトの進歩と機械の普及で定着したワープロの特集。最新の機能のチェックと，ワープロの可能性についての対談など。――編集部，ASCII，11月号，217-243pp.

▶CD-ROMって，こんなに便利
アイシーエムから発売されているドライブ「CD-500E」と，それにバンドルされている検索機能付きエディタを検証する。――志村拓，ASCII，11月号，245-248pp.

▶ACCIDENTAL EMPIRES
米国のコンピュータ界の天才たちの人間模様を描く「コンピュータ帝国の興亡」より，一部を抜粋して掲載。IBMをめぐるトピックに触れる。――Robert X.Cringley，ASCII，11月号，249-256pp.

▶ことば遊び・コンピュータ
文を自動生成するプログラム「〇子」。コンピュータならではのイリーガルな文章が笑いを誘う。――ホーテンス・S・エンドウ，ASCII，11月号，337-341pp.

▶TBN・SHOW
千葉・幕張で開催の「第30回アミューズメントマシンショー」。今回は最終日の一般公開で，従来と雰囲気の違うイベントに。――魚住晶，ASCII，11月号，362-363pp.

▶バカパパのモノを買い物
パソコン周辺グッズの永遠の定番，フロッピーケースを再び取り上げる。ほかには２進数表示の腕時計などが登場。――バカパパ，ASCII，11月号，366-367pp.

▶なんでも相談室
マークシートは鉛筆でないと読めないが，競馬はボールペンでもOK。その理由を解説。そのほか，MS-DOS関連の質問など。――編集部，ASCII，11月号，374-377pp.

▶欧州ハイテク事情
経済・社会両面で混乱の続くロシアだが，そこでも確実に普及しているコンピュータについて，最近のニュースを解説。――菊地薫，ASCII，11月号，384-385pp.

▶ラッキー！ハッピー！オッケー！
BBS上で悪口をいわれた場合，名誉毀損として訴えることはできるのか，賠償責任者は誰か？　という質問に弁護士が答える。――編集部，ASCII，11月号，388p.

▶マイコンからMy Computerへ
創刊15周年記念企画。ジャストシステム社長浮川和宣氏などにソフトビジネスの現状などについて聞く。――編集部，My Computer Magazine，11月号，64-75pp.

▶データショウ'92レポート
「人にやさしい情報環境の創造」をテーマに東京・晴海で行われたデータショウのレポート。――電脳事務，My Computer Magazine，11月号，99-104pp.

▶ソフトメーカーが創った新時代のテーマパーク
ナムコ「ワンダーエッグ」のレポート第２弾。アミューズメントのメカトロニクス技術を紹介。――野沢潤一郎，My Computer Magazine，11月号，114-117pp.

▶NEWなリリース　日本人をハダカにする!?
日本人の体格向上で実情にそぐわなくなったJIS規格。新規格作成のための人体計測キャラバンを取材。――編集部，My Computer Magazine，11月号，132-135pp.

▶PC実験室
ディスプレイフィルタを検証。比較的安価な３製品の，紫外線透過や静電気，映り込みなどをみる。――石川至知，My Computer Magazine，11月号，148-153pp.

▶ビジネスマンのための情報管理術
シャープハイパー電子システム手帳DB-Zの活用講座。通信カードでパソコンとの直接通信を試みる。――塚田洋一，My Computer Magazine，11月号，211-215pp.

▶マイコン考古学
「パソコンサンデー」の宮永好道が贈るマイコン歴史学。OSの成り立ちとMS-DOS普及までの経緯を語る。――宮永好道，My Computer Magazine，11月号，275-278pp.

▶通信コストをダウンサイジング！
パソコン通信利用者にとって頭の痛い電話代。各種の割引サービスにより，安くあげる方法を研究。――高橋雄一，My Computer Magazine，11月号，279-286pp.

▶データショウ'92
東京・晴海のデータショウのレポート。――編集部，I/O，11月号，104-105pp.

▶スーパーコンピューティング入門
数学パズルシリーズの第３回。巡回セールスマンの課題を例にとり，P＝NP問題と呼ばれる現代数学の壁について触れる。――林智雄，I/O，11月号，156-159pp.

▶U.S.A.Rupo
米国のコンピュータ最前線を知るページ。１回目はバーチャルリアリティへの各企業の取り組みを紹介。――ティム・スキャネル，月刊PC，11月号，241-248pp.

MZシリーズ

MZ-1500(BASIC MZ-5Z001)

▶カドマソア
ダライアス風シューティング。――FROG，マイコンBASIC Magazine，11月号，119-121pp.

MZ-2500(BASIC-M25)

▶Border
国境線までつっぱしれ！　左右にトラックを操り，頭上からの攻撃と地上の岩をよける。――アダモ，マイコンBASIC Magazine，11月号，122-124pp.

▶誌上公開質問状
読者の疑問にメーカーが回答する。MZ-2521につなぐことができるプリンターやケーブルを紹介。――I，マイコンBASIC Magazine，11月号，150-151pp.

X1/turbo/Z

X1シリーズ

▶FRUIT VILLAGE
たわわに実ったリンゴを，アップル星人から守れ！　アクションゲーム。――もったんSOFT，マイコンBASIC Magazine，11月号，150-151pp.

▶影の追跡
編集部をクビになった影さんの逆恨み。パソコン破壊で報復だ！　編集者のあなたは暗闇で見えない影さんか

参考文献

I/O　工学社
ASCII　アスキー
月刊PC　ソフトバンク
コンプティーク　角川書店
C MAGAZINE　ソフトバンク
テクノポリス　徳間書店
POPCOM　小学館
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

らパソコンを守らなくてはならない。――天野絢也，マイコンBASIC Magazine，11月号，152-154pp.

X68000

▶GAMING WORLD
　アクションゲーム「ストライダー飛竜」や，戦車による戦闘シミュレーション「スクエアリゾート」を紹介。――編集部，テクノポリス，11月号，44，47pp.

▶SOFT EXPRESS
　アクションゲーム「バーンウェルト」。――編集部，コンプティーク，11月号，61，65pp.

▶How To Win
「三國志III」を年代別に攻略。――編集部，コンプティーク，11月号，108-111pp.

▶SUPER NEW SOFT
　ズームの次回作，F1レースゲーム「オーバーテイク」。――編集部，LOGIN，19号，14-33pp.

▶最新ゲーム徹底解剖!!
「ポピュラスII」を基本から攻略。各種のイベントやアイテムを使って楽しもう。ほかに，PCエンジンからの移植「ネクタリス」や，シミュレーション「シュートレンジ」を紹介。――編集部，LOGIN，19号，130-157pp.

▶X68000新聞
　新作「オーバーテイク」で燃えているズームの開発部隊を取材。かわいくって楽しいゲーム「エトワールプリンセス」。DōGA CGAシステムのバージョンアップのお知らせ。――編集部，LOGIN，19号，218-221pp.

▶最新ゲーム徹底解剖!!
「オーバーテイク」。月面が舞台の戦闘シミュレーション「ネクタリス」は各ステージ別攻略ポイントを解説。「ポピュラスII」徹底解剖の最終回，カスタムモードをプレイしよう！　――編集部，LOGIN，20号，120-145pp.

▶X68000新聞
「バーンウェルト」。「CHART PRO-68K」。――編集部，LOGIN，20号，236-239pp.

▶PIKIN
　エイリアンの侵略を阻止しろ！　壁に跳ね返るレーザー兵器を使ったインベーダーゲーム。――梅津　毅，マイコンBASIC Magazine，11月号，155-156pp.

▶飛昇蝶
　ダメージを受けないように蝶を操り，女蜂を回収。アクションゲーム。――千吉良賢一，マイコンBASIC Magazine，11月号，157-159pp.

▶マリ夫の冒険パート3
　ブロック上を飛び回って鍵を取る。でも一度マリ夫がジャンプしたところは穴が空いてしまうのだ。――高橋秀之，マイコンBASIC Magazine，11月号，160-162pp.

▶MISTY BLUE　～Hold me tonight～
　エニックスのゲームミュージックプログラム。要NAGDRV＋MT32。――佐々木嗣朋，マイコンBASIC Magazine，11月号，175-176pp.

▶誌上公開質問状
　X68000 ACEに増設できるメモリは？　PROシリーズにCompactのキーボードは繋がるか？　などの質問に回答。――多田太郎，マイコンBASIC Magazine，11月号，177p.

▶FREE SOFTWARE INDEX
　大手BBSにアップロードされた便利なツールやゲームを紹介。X68000用のHuman68k拡張ユーティリティ，アイコン集など。――編集部，ASCII，11月号，401-407pp.

▶AV STRASSE
「X68000ふたたび」と題し，「スーパー金魚すくい」の作者である石田博也さんに話を聞く。X68000の将来は明るいか？　――編集部，ASCII，11月号，323-324pp.

▶なんでもQ&A
　X68000Compact用の内蔵ハードディスクの仕様と「Communication SX-68K」の紹介。CHART PRO68KでBUSINESS PRO-68Kのカルクシートを転用できるか？　の問いに答える。――シャープAVCシステム事業推進室，My Computer Magazine，11月号，264-265pp.

▶フルカラーBMPファイル・ローダ
　X68000用のBMPファイルローダ。2/16/256/フルカラーの画像ファイルも表示可能。――伊藤ゆう，I/O，11月号，152-153pp.

▶月刊PCが選ぶ，92年ベストパソコン　ノミネート編
　ハード，ソフトのベストバイを決定する特集。今月はX68000XVIを含め20種のパソコンをノミネート。――荻窪圭，月刊PC，11月号，146-176pp.

▶ゲームで遊ぶ30万円台パソコン
　人気パソコンのゲーム機としての特性や得意なジャンルを探る。X68000のほか，PC-9801，DOS/V，Macintoshなどを検証。――編集部，月刊PC，11月号，225-231pp.

▶GCCで学ぶX68Kゲームプログラミング
　ゲーム作成の連載。今月はパチンコを題材にC言語でシミュレータを作ってみる。パチンコ大当たりの確率論つき。――吉野智興，C MAGAZINE，11月号，129-134pp.

ポケコン

PC-E500

▶サイキックパワー
　放っておくと落ちていくドットを，君のサイキック・パワーで引き上げるのだ！　――Oogezu Soft.，マイコンBASIC Magazine，11月号，164p.

▶モグラシバキゲームだ！
　タイムオーバーまでに何匹のモグラをしばけるかな？――大島孝広，マイコンBASIC Magazine，11月号，165p.

新刊書案内

プログラマの妻たち
ビレッジセンター編
ビレッジセンター出版局刊
☎0424(88)9421
B5判　254ページ
1,300円(税込)

　とにもかくにもユニークな試みであることに違いはない。パソコンのプログラマ，特に職業プログラマであると同時に趣味のプログラマであり，フリークでありマニアであり，放っておくと1日中でもパソコンに向かっているような人たちの，その奥さんが自分の旦那について書いた本なのである。だいたい奥さんがいるような年代であるから，パソコン業界に早くからかかわってきた人たちが多く，その筋の有名人も「電脳騒乱節」の中村正三郎からDOS/Vの西川和久など揃っており，なおかつ夫婦の写真まで入っていたりして，パソコンマニアにはたまらない内容なのだ。

　原則として，ふたりの出会い，プロポーズ，旦那の性格，生活，趣味といったものがちりばめられており，文体，表現などそれぞれ個性に富んでいる。さすがにプロの文章ではないから，前から順番に読んでいくと飽きるが，名前を知っている人から暇つぶしのつもりで読んでいけば，楽しめることは保証しよう。

　プログラマはどこで奥さんを見つけるのか。学生時代に見つけた人もいるし，就職後，仕事に関係のあるところ(つまり，コンピュータ業界)で見つけた人もいる。

　こういう本に出てくるくらいだから，旦那は一応まともな生活を送っているのだろう，などと思っているとそれは間違い。読者の想像どおり，旦那はパソコン漬けで，女性の目から見たパソコンマニアの生活を描いた書，といっていいくらいだ。パソコン漬けの生活をしていても，部屋中がジャンクとフロッピーディスクとパソコンにあふれていても，ちゃんと結婚はできるのである。しかも，けっこう美人と。

　前書きで，中村正三郎が「バカヤロー」と書いている。先日，彼と会ったときに「読んだよ」といったら「俺はまだ怖くて読んでない」といっていた。とまあ，そういう本なのである。　(K)

踊るコンピュータ
東野　司著
ジャストシステム刊
A5判　254ページ
1,600円(税込)

　SF作家・東野司氏による書き下ろし読み物。日記形式，会話形式などさまざまなかたちの生き生きした文章に加えて，イラストあり，アヤシイ写真あり，の楽しさである。ページのレイアウトでも，ちゃんと「遊んで」いる。「キーボードの中に触ったことのないキーがある」「帰るアドレスがない」などなど……この本に出てくる言葉にドキッとするあなたは，パソコン病のおそれが濃厚。

　パソコンをアイしちゃってる人，コンチクショウと思ってる人，機械アレルギーの人，パソコンなしには生きてゆけない人……そんな人々に贈るパスティーシュ小説集のような一冊である。

コンピュータ社会と漢字
マーシャル・アンガー著
奥村睦世訳
サイマル出版会刊
☎03(3582)4221
B5判　299ページ
3,500円(税込)

　著者は米国・メリーランド大学教授の言語学者である。本書では，日本語表記を分析，理解したうえで，そのコンピュータ処理における問題の巨大さを提示し，解決方法を提案している。

　コンピュータの普及により，日本語の表記は変わるのか，果たして日本人は漢字を捨てることができるのか。この問題は，単なる処理の効率化ということでは片づかない大きな文化的問題をはらんでいる。日本語表記が「日常使用の伝達手段」ではない第三者である著者の，学問的立場からの視点で述べられた提案を，私たち日本人自身は，一度は耳を傾け，考えてみる必要があるようだ。

QUESTION and ANSWER

Q **X68000をタイマでセットした時間に電源ONさせるにはどうしたらいいのでしょうか？ 具体的に手順を追って説明お願いします（X68000を目覚まし時計にしようと思っているので)。タイマONできれば，AUTOEXEC.BATから音楽プログラムを流したいのですけど，そのバッチファイルの作り方も教えてください。曜日によって曲を変えるとか，OPMデータをどうやって演奏させるかなど。使用機種はX68000XVIです。**

静岡県 間淵 繁紀

A まず本体付属のSX-WINDOWが収められたシステムディスクでX68000を起動してください。起動したらコントロール.Xを実行します。コントロール.Xを実行するには，コントロール.Xのアイコンをダブルクリックする方法と，アクセサリのポップアップメニューから選択して起動する方法があります。とりあえずアクセサリで選択して起動する方法を説明しましょう。X68000アイコンにマウスカーソルを重ねて，右クリックでポップアップメニューを開いてください。メニューにあるコントロールパネルを選択します。すると設定してあるパスを検索してコントロール.Xを自動実行してくれます。

コントロール.Xを実行すると，コントロールパネルウィンドウがオープンします。ウィンドウ内にはディスプレイ，マウスやキーボードのアイコンが並んでいますね。この中の時計アイコンにマウスカーソルを重ねてダブルクリックしてください。新たにウィンドウが開きました。操作はややこしいところがあるので，本体付属のマニュアルでコントロールパネルの操作説明をよく読んでX68000をTIMER ONにする時間を設定してください。本体全面のTIMERランプが赤く点灯すれば設定できています。

設定した時刻に電源ONになるか不安なら，一度現在時刻の2，3分後にタイマONを設定して試しておくといいでしょう。設定したらX68000の電源をOFFにします。そして待ちます。……ちゃんと設定した時刻に電源が入りましたか？ うまくいかない場合は，

・時刻の設定が正しかったか
・本体全面のTIMERランプが赤く点灯していたか

チェックしてください。

ちなみにタイマで電源をONにすると，POWERランプとTIMERランプが点滅しますが，これは故障ではないので安心してくださいね。

それとOPMファイルを鳴らすには，

A>COPY MORNING.OPM OPM

のように，OPMファイルをシステム予約ファイル名OPMにコピーします。最近はZ-MUSICシステムで演奏するものがOh!X LIVEに多数掲載されていますが，それらも同様に，

A>COPY MORNING.ZMS OPM

のようにすれば演奏することができます。ただしOPMDRVn.XやZ-MUSICがシステムに組み込まれていないと，OPMというファイルがカレントディレクリに作成されるだけですから注意してください。

さらに曜日によって違う曲を演奏させたいとの質問ですね。ここで問題になるのはバッチファイル中で曜日を得るコマンドがないという点です。曜日を知るためのプログラムを自分で作成するとか，他人の作ったものをもらうとかして，用意する必要があります。昔，質問箱でも曜日によって違う曲を演奏するために必要なサンプルプログラムを紹介したことがあります。リストも短いしこの手の質問はよく送られてくるので再度説明することにしましょう。

まずAUTOEXEC.BAT中での曜日の扱い方です。これは日曜日を0，月曜日を1，火曜日を2……土曜日を6というように，曜日を数字に対応させることにします。曜日を調べる外部コマンドを作成したら，曜日に対応する数値を外部コマンドの終了コードとします。外部コマンドの終了コードとして，ユーザーは0～255を自由に使うことができます。

それ以外のエラーコードにどんなものがあるか興味を持った方は，Human68kユーザーズマニュアルのCOMMAND.Xの説明を見てください。ひとつ紹介すると，たとえばCOMMAND.Xの内部コマンド中でファイルが見つからなかった場合は，$0503(1283)というエラーコードを返すようになっています。

なぜ曜日を数字で表すかというと，バッチファイル中でIFコマンドのEXITCODEオプションで曜日を判定するためです。

では早速先ほど説明した仕様に基づいた外部コマンドを作成しましょう。ここではもっとも簡単に，C言語のメイン関数の戻り値に曜日に対応する数字を返すようにしました（リスト1)。リストを入力したら，

A> CC /Y DAY.C

としてコンパイルしてください。スイッチのYは必ず大文字です。少しばかりWarningが表示されますが気にしなくていいです。無事コンパイルが終わると，DAY.Xというファイルができていると思います。コマンドラインから，

A> DAY

とやっても表面上はなにも起きませんが，バッチファイル中でDAYコマンドを実行してEXITCODEを調べてやると，DAY.Xが曜日に対応したコードを返してるのがわかると思います。

ついでにリスト2に音楽演奏させる場合のバッチファイルのサンプルを紹介しておきましょう。演奏させるファイル名（SUN.OPMなど）は各自変更してください。

このようにC言語を使えばバッチ処理用のコマンドを簡単に作ることができます。ではC言語がない場合はできないのでしょ

リスト1

```
============  day.c  =============
 1: #include <stdio.h>
 2: #include <doslib.h>
 3:
 4: main()
 5: {
 6:      int date;
 7:      char day;
 8:
 9:      date = GETDATE();
10:      day =  date >> 16;
11:
12:      EXIT2(day);
13: }
```

リスト2

```
==============  test.bat  ===============
 1: echo off
 2: day
 3: if exitcode 0 copy sun.opm opm
 4: if exitcode 1 copy mon.opm opm
 5: if exitcode 2 copy tue.opm opm
 6: if exitcode 3 copy wed.opm opm
 7: if exitcode 4 copy thu.opm opm
 8: if exitcode 5 copy fri.opm opm
 9: if exitcode 6 copy sat.opm opm
10: echo on
```

うか？　いえ，多少複雑ですが，バッチを駆使すればなんとかなります。まず，

A>ED CR

としてエディタ起動後，Ctrl-V，リターン，ESC，Eの順にキーを押します。次に“現在の日付は.BAT”という変な名前のバッチファイルを作ります。内容は，

COPY %2 OPM

という1行だけのものです。そして，AUTOEXEC.BATの末尾に，

DATE <CR> MUSIC.BAT
MUSIC

を加えてください。

あらかじめ，曜日ごとに鳴らしたい曲を“(月)”，“(火)”という名前（拡張子なし）で用意しておくのがミソです。

X68000PROIIを持っています。RAMを増設したいのですが，現在PIO-6BE1-Aの内蔵RAMを増設して2Mバイトになっています。次に増設するには，どのRAMをつけたらいいのでしょうか？

広告を見るとスロット用と書いてありますが，スロットにはMIDI/SCSI/ビデオボードがついていて，全部使っています。スロットにはつけることができません。どうしたらいいのでしょう？　福島県　高田　英夫

質問を読むとMIDI/SCSI/ビデオボードが拡張スロットに差さっているということですね。X68000PRO IIなら，拡張スロットは4つありますからひとつスロットが残ってるじゃないか。こう思う人もいることでしょう。でも，ビデオボードは2スロットを占有するのでした。

本体に内蔵可能なRAMも増設されていますので，もはやRAM増設はスロットを使うしかありません。現在スロットをふさいでいるボードのうち，もっとも使用頻度の少ないものを抜いてRAMボードを取りつけるしかないようです。SCSIは外せないでしょうから，MIDIかビデオボードか……，ということになるでしょう。使用頻度が少ないからといっても使うたびに背面に手を回してボードを抜き差しするのは面倒です。考えてみれば，わざわざ質問箱にハガキを送ってくるほどだから，どのボードもよく使うんでしょうね。

ビデオボードについている端子を見たことがあるでしょうか？　これがとても少ないのです。調べてみるとビデオボードは電源を取るためだけに拡張スロットを占有しているのだそうです。ですから電源をほかから取れば，ビデオボードを拡張スロットに差し込む必要はなくなります。2年ほど前になりますが，DōGACGアニメーション講座の連載の中で，ビデオユニットの電源をX68000本体のカラーイメージユニット用のイメージ入力端子から取るようにする製作記事が掲載されました。

製作は簡単なのですが（製作というほどのものでもない），質問箱に与えられているページ数で説明するのは苦しいです。掲載号は1990年10月号ですので，手元にあるようでしたら参考にしてみてください。バックナンバーの入手は困難かもしれません。

自作はちょっと……というのであれば，割高にはなりますが，パソコンショップで販売されているビデオボード外付けユニットの購入をお勧めします。本誌1992年3月号前後の計測技研/BASIC HOUSE（☎0286(22)9811）の広告には，ビデオボードケース（9,800円）という商品名で販売されていました。現在でも扱っているかどうかわかりませんが，一度電話で問い合わせてみたらいかがでしょうか。

高田さんの場合はスロットが4つあるのでまだいいのですが，2スロットしかないEXPERTやXVIユーザーはビデオボードだけですべてのスロットがふさがれてしまいます。悪名高いビデオボードというのもうなずけます。　（影山　裕昭）

Z-MUSICシステムのMMLを使ってCM-64の初期化をする場合は，

.ROLAND_EXCLUSIVE　$10，$16
{$7F，$00，$00，$00}

で行えるそうですがSC-55でもそのような機能はありますか。　埼玉県　阿蓋　達也

本誌1992年10月号82ページでCM-64/32Lの初期化について書かれていますが，CM-64/32Lではアドレス$7F，$00，$00にデータを送信すると楽器の設定が初期化されます。10月号の記事では，

.ROLAND_EXCLUSIVE 16,22
{$7F，$00，$00}

となっていますが，これではアドレスを指定しただけでデータを送信したことにはなりませんので初期化は実行されません。もう1個$00を書いて（別に$00でなくてもいいのですが），

.ROLAND_EXCLUSIVE 16,22
{$7F，$00，$00，$00}

のようにすべきです。

さて，もちろんこの初期化機能を持ったアドレスはSC-55シリーズ（GSモジュール）にも存在します。SC-55のマニュアル78ページの下段を参照してください。

「システムをリセットし～初期化します」という説明のアドレス$40，$00，$7Fが発見できることでしょう。ここへCM-64/32L同様にデータを送信してやれば初期化が行えるわけです。具体的には，

.ROLAND_EXCLUSIVE $10,$42
={$40，$00，$7F，$00}

とします。

こういった機能はヤマハTG100にもあります。TG100では同機のマニュアル94ページにあるとおりのメッセージを送信してやることで実現できます。

.EXCLUSIVE　{$43，$10，$27，$30，$35，$06，$00，$15}

となります。

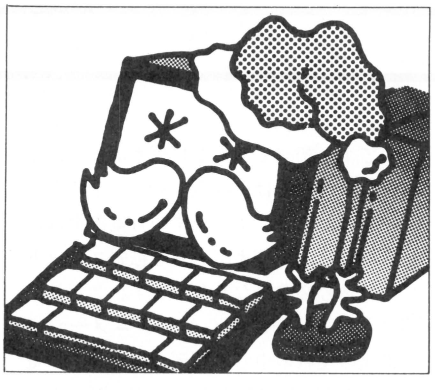
こういった初期化メッセージは特定のMIDI楽器を用いた演奏データを作成するときは必ず行いましょう。行わないと前に演奏していた曲の設定が残ることになり希望どおりの演奏がなされないことがあります。　（西川　善司）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してくださいね。

宛先：〒108　東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部
Oh! X編集部「Oh! X質問箱」係

STUDIO X ON AIR



FROM READERS TO THE EDITOR

今年も残りはあとわずか。さあ，やり残したことを一気にやり尽くすんだ！ 12月は，忘年会やらクリスマスやら，イベントだって盛りだくさん。「寒いから」なんていって，おうちにとじこもってちゃダメですよ。

◆ヨ，ヨンセンマンだあー。実は私の就職志望先のなかに某CG制作会社があったのですが，そこのモニタにヨンセンマンがぐにょぐにょ動き回っていたんですよ。で，かんじんの入社試験の結果はといいますと……。くそー，ヨンセンマンなんかキライだー(さかうらみ)。

嵯峨　進(23)秋田県

う～ん，希望プレゼントがヨンセンマンじゃなくてよかった(やつあたりするために欲しい，なんていわれちゃったら……ちょっとフクザツ)。

◆シャープシステムサービスに就職試験を受けに行き，みごとに撃墜されてしまった。もっと「X68000を使っています」とアピールしておけばよかったなあ。シャープシステムサービスに合格した人，これからシャープのため，私たちのためにがんばってください。やっぱり，成績かなあ。う～ん!?　水谷　国宏(18)滋賀県

今年の就職は厳しいそうですね。求人も年によって波があるけれど，人生も楽あれば苦あり，大丈夫ですよ，ねっ。

◆最近，人生に疑問を感じています。私の人生って，なんなんでしょう？

山岸　稔(19)埼玉県

だから，楽あれば苦あり，いろいろあるんですよ，きっと。いろんなことにトライして，考えてみてね。

◆今度発売される「デスブレイド」，実は私が某データイーストに勤めているときに，初めてメインでグラフィックを担当させていただいた作品です。すでにあの頃のグラフィックの者の大半は別の職場へと「とらばーゆ」し，企画者も辞めてしまったようです(人の出入りが激しい業界ですね)。今見るとドット絵の見せ方もまだまだ甘く，恥ずかしい気もしますが，X68000に移植されるのはうれしいです。

橋本　和典(25)東京都

生き別れた子供が成長して戻ってきて再会，なんていったらおおげさかな？　でも，「自分が使ってるマシン」への移植ですから，特にうれしいのでは……。発売が楽しみですね。

◆10月号のなかよし姉弟を読んだ。おや？　こんなところに「SHAKE HIP」と書いてある。岡村さんは米米CLUBのファンですか？　(ボクもこの服欲しいなー)　木村　亮(19)静岡県

ファンの多い「なかよし姉弟」。電子ちゃんとともに，「特集」とか「別冊付録」への希望の声も。岡村さん，これからもがんばってくださいね。

◆9月下旬から10月初めまでの2週間，教育実習に行ってまいりました。2名ほど，Oh!Xの読者が生徒にいて，ときどきパソコンの話で盛り上がりました。毎日3時間睡眠だったけど，終わってみるとあっという間でした。あぁ，これでやっとゲームができるぅ(2週間，一度もX68000に触れなかった)。　岩瀬　貴代美(20)福岡県

で，そのつづきの生活が，下の投稿イラストにあるのですね。うんうん。まあ，なにはともあれ，おつかれさまでした。

◆10月号の「Oh!X LIVE SPECIAL」には驚いたが，それ以上に「よいこのSX-WINDOW講座」にはビビりました。リスト2のあの文章には，あのCDを聴かないとわからないネタがあるし，以前の「SHIFT BREAK」にもあれにハマっている人がいたようだし，ここらへんで特集でも組んでみたらどうでしょうか？　ディスクでもつけて，CGやMUSIC，さらにDōGAによる変身シーンでも入れれば，売り上げ倍増まちがいなしでしょう。　由岐中　康司(21)神奈川県

セーラームーンファンは，現在，着実に増え続けている模様です。

◆セーラームーンのカレンダーを注文してしまった。　井戸　直樹(22)岐阜県

あああ……とうとう，こんな人も……。

◆あ～。みんなセーラームーンにはまってる！

杉山　隆一(18)埼玉県

そういうアナタも，もしかして……？

◆10月号の「LIVE SPECIAL」の美少女戦士セーラームーンを娘に見せたら，ぜひ聴きたいとのこと(娘は毎週見ていてファンです)でしたが，X1用なので，X68000用に変換できないか思案中です。　小林　賢(36)埼玉県

セーラームーンファンっていうと，最近はなぜか，ある一定の年齢層の男性が思い浮かんできてしようがなかったのですが，こういうお話を聞くと，ちょっと安心(?)……(ナゼだろう)。それはさておき，これを機会に，お嬢さんと一緒にX68000を使ってみてはいかがでしょう？　パソコンの「英才教育」も，いいかも。

◆突然，うちのカミさんが「上海」に狂ったので，まず「信州」をやらせてあげた。おまけに「スーパー上海ドラゴンズアイ」を買ってあげた。これで我が家のX68000の株も上がったに違いない。冬のボーナスでハードディスクが買えるかもしれない。　木越　英夫(33)愛知県

パソコンユーザー増加計画の第一歩は家庭内から！　周辺機器の充実で快適なパソコン生活を！　……でも，そのうちに，熱中した奥さんにX68000をとられちゃったりして……。そしたら，もう1台買って，一緒にOh!X読んでくださいね。

◆進藤さんが無茶な注文をつけるので，何かMIDIを買ってこようかと思います(家を建て替えるそうなので，今はやめとこうと思ってたのに)。ローランドさんは彼に菓子折りでも持って

▲岡村　直也　兵庫県
2コマ目のヌイグルミみたいな姉ちゃんがカワイイ！　でも，姉ちゃんは「太るよりもウイルスのほうがまし！」っていうかも……。

▲岩瀬　貴代美　福岡県
岩瀬さんは1週間もゲーセンに通いづめとのこと。2週間の禁ゲーム生活の反動かな？　学祭やクリスマスも楽しんでね。

いってください。ところで，マニュアルが初心者向けの音源て，どれでしょう？　いくら性能がよくても，使いこなせなかったらどうしようかと思ってます。店の人に聞こうにも，MIDI＝ミュージ郎しか知らないんだもん，某店は……。大阪まで出かけようかな……。

雄井　理恵(25)和歌山県

某氏によると，SC-55のマニュアルは結構いいよ，とのことですよ。それにしても，初心者にとっては，マニュアルのわかりやすさって，とても大事なことですよね。でも，もうひとつ方法があります。それは，詳しい人とお友達になって，その人を「人間マニュアル」にしてしまうことです。知ってることをお互いに教え合えば，バージョンアップ(人間の，ね)もカンタンです。

◆10月号の「STUDIO X」に「S端子が欲しい」という話がありましたが，私は日本ビスコム の「S-RGB(21P)プロセッサ」をCZ-600Dにつないでいます。すると，ディスプレイTVが「高精細モニタ」に早変わりします。そこいらの低画質大画面TVなど目じゃありません。しかし，プロセッサが高い(38,800円)のと，多少の改造が必要(21ピンと15ピンの同期信号の違いのため)なので，マニア向けかもしれません。というわけで，私のディスプレイTVへの要望は，「Y/C分離に三次元フィルタを！」です(シャープのビデオ「VC-BS500/400」にも搭載されています)。
P.S.　CRTフィルタ「BF68PRO」をつけると，黒が締まって色がきれいになります(いわゆるラベンダーマスク)。　植木　正幸(23)神奈川県

改造が必要，というのはすべての人にオススメ，とはいきませんが，こういう手もあるのですね。さすが，X68000ユーザー(？)。

◆ひと足早いサクラサク。大学院に合格しました。とりあえず，今後2年間の行く先が決まり，ほっとしています。ところで，10月号の表紙の生き物，これ何でしょうかね。思うに，これは体を半分に切断しても両方とも再生するというプラナリアではないでしょうか。

松永　貴輝(22)愛知県

◆10月号の表紙は何？　水玉ですか？

早川　敏志(17)宮城県

◆これから，表紙絵に「題」をつけてもらいたいですね。10月号は金色のおたまじゃくしなのでしょうか？　田中　拓弥(19)愛知県

◆10月号の表紙絵は，いったいなんだろう？　水滴がもし生きていたとしたら……とか。きっと，おたまじゃくしなんでしょうね！

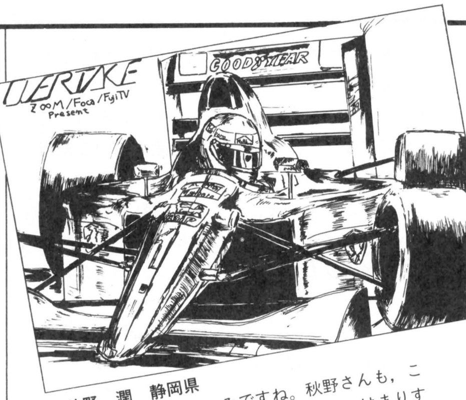
宮沢　毅(22)長野県

◆10月号の表紙って，なんかアヤシイ……。最近，表紙がブキミになってきたなぁ。

服部　直幸(19)福岡県

◆10月号の表紙はすごいですね！　なんかツルツルで……。自分はこういうCGはとても好きです。　田鍋　弘司(18)広島県

◆10月号の表紙は，どっちが上か下か悩んでしまうCGですね。本当は，上下が逆じゃないんですか？　(いや，まてよ。やっぱりこれでいーのかな。うーむ)　佐藤　一秀(35)愛知県



▲秋野　潤　静岡県
オーバーテイク，楽しみですね。秋野さんも，このためにMIDIを購入予定とのこと。でも，はまりすぎてイラスト投稿やめちゃったりしないでね。

▲鈴木　貴久　静岡県
これは，シャープのペンワープロWD-A750を使用してのイラストだそうです。うん，なかなか味があってよいなあ。

◆10月号の表紙のモデルは，もしかして，昔テレビでやっていた，ヒューヒュー，ポーポーとかいう名前のオバケではないでしょうか？　実は新型のツインビーとか？　指が4本ということは，ミッキーマウスのパワードスーツ量産型とか？　それともバグの大群？　あー気になる。ぜひ正体を教えてください。逆さに見たほうが自然に見えるのは歳のせいでしょうか？

谷本　俊康(26)東京都

ほかにも，イラスト入りのハガキとか，「きれい」「ブキミでよかった」「洋梨だ！」などなど，いろいろなご感想をいただきました。人それぞれ感じ方が違うのっておもしろいですね。作者の方は，いまごろニンマリしていらっしゃるかも。

◆裸眼立体視がうまくできない人にコツを少し。
平行法：絵をつき抜けて遠くを見る
交差法：絵と目の中間の空間を見る
というつもりでやると，うまくいくと思います。

黒田　隆之(22)大阪府

◆立体視！　一時期，熱中しました。一度できるようになると，もうヤミツキになります。当時，大学の図書館で立体視図を探しまくりました(生物，化学に多い)。意外な利用法としては，間違いさがしクイズに使えるということです。異なっている部分がちらつきます(マガ○ンの最後に載ってるヤツなど楽勝)。

杉村　謙一郎(24)静岡県

なるほど，そんな利用法が……というわけで，さっそく試してみた担当者ですが，間違いさがしよりも立体視の練習のほうが時間かかっちゃいました(でも，これからは楽勝さっ！)。

◆某A大学の学校案内のパンフレットを眺めていると，学生の寝室の机の上になにやら見なれたものが……。おお，よく見るとそれはX68000PRO(II？)ではないですか。シャープは防衛庁の管轄にまでシェアを広げていたのですね。さすが目のつけどころが，以下略。

国部　恭司(18)佐賀県

よく気がつきましたね。さすが目のつけどころが(？)。もしかして，そうやってじわじわとX68000の宣伝をしてるのだったりして……。ついでに，さりげなくOh!Xも置いててくれたらよかったのになぁ。

◆「チャーハン」と「焼飯」と「ピラフ」の違いを教えてください。　横田　耕一(22)愛知県

岩波の国語辞典によると，
チャーハン【炒飯】→やきめし。▷中国語。
やきめし【焼(き)飯】①飯を油でいためて作った料理。いりめし。チャーハン。②握り飯を火であぶったもの。
ピラフ　飯に肉やえびを入れ，いためた料理。▷pilaf
とのことですが……。どう違うの？

◆皆さん，「島野路子」と書いてなんと読み，これが何かご存じですか。「しまのみちこ」と読んで女性の名前という人がいるかもしれません。間違いとはいいませんが，実は「しまのじこ」という鳥の名前なのです。ちなみにワープロの変換では「島の事故」になってしまいますが，日本語というのは難しいですね。

藤原　彰人(22)岡山県

珍しい名前の人などは，おかしな変換をされてしまって腹を立てることもあるのではないでしょうか。鳥はきっと怒らないだろうけどね。

◆ASK68Kで「おすまん」を変換すると，「オスマン帝国」と，勝手に「帝国」の2字が入るのは大きなお世話だと思う。こんなことを知っているのはイラン史のレポートを書くハメになったからなのだが(もう提出したよ♪)，それにしても，なぜ「オスマン」だけおまけつきで変換されるのだ。　矢野　啓介(19)北海道

ほかにも，おまけつきのことばがあるかもしれませんね。でも「おまけ」って聞いて，なんか得した気になるお馬鹿さんは，やっぱり私だけ……？

◆メガネ屋のX68000についてですが，僕は6年くらい前に見ました。といっても，いまのように箱に入ってはおらず，カメラやディスプレイはむきだしで(キーボードと本体はいちおう，うしろに隠してあった)，まだ「試作段階」という感じでしたが……。そこでの話。初めてX68000に触れた私は，「おお，これがトラックボールか」

と，ボール部分を思いきり，ズドン……。このあとは……ぜひご自分のマシンでお試しください。合掌。

追伸　(で)さん，今度復活祭してください。

坪田　雅己(17)広島県

トラックボールとのおつきあいは，「破壊」から始まったのですか……。そのときからトラックボールの「霊」に憑かれているあなたは，きっともう一生X68000から離れられません。大事にしてね。

◆夏休みのバイトも終わり，これでハードディスクやCコンパイラが買えます。「遺跡発掘調査」と聞こえはよいが「土方」と同じようなことを1カ月もやってきたので，腰がとても痛いのです。しかも再試が山ほどひかえているのに……(ちなみに1コ，スゴイ石器を見つけた)。

坂井　国彦(20)静岡県

す，すごい！　どんな石器を見つけたんでしょう。欲しいものも買えるし，充実したバイト生活でしたね。きっと腕力もついたでしょうし……。

◆ソフトの発売日は，遅れて当たり前といった風潮があるようだが，一般の常識に当てはめると，すごくおかしくないですか？(だって普通，仕事の納期が遅れるって大変なことじゃないですか)

左　拳(23)愛知県

そうですよね！(と，思わず声が大きくなる)　予定どおりに発売してくれないと，雑誌の編集日程が……。でもまあ，よりよいものを作るために，いろいろ事情があるのでしょうね。メーカーの方々，がんばってくださいね。

◆CDを分解して掃除する僕を，友人は変な目で見る。なぜだ，なぜなんだ～。

二口　康宏(20)福井県

まさか，テレビとかビデオとかパソコンとか掃除機とか洗濯機とかエアコンとかトースターとか炊飯器とかFAXとかコピー機とか……なんでも分解して掃除してるんでは？　(ウチのもしてほしいなぁ……)

◆日本へ帰りたい……。谷口　博一(26)大阪府

外国でのお仕事は，きっといろいろ大変なのでしょうね。でも，元気を出して，インドネシアならではのユニークな体験記とか，お待ちしてますよ。ねっ。

◆ぐしぐし(©岩瀬貴代美)って書くと載るって本当ですか？

各務　剛二(19)愛知県

嘘です。あれ？　でも，このハガキ載っちゃったなあ。ま，いいか。特別にね。

◆ハードディスクを買うためにバイトを始めました。メモリもコプロも欲しいし，バイクと新しいスキーウェアと服と，それから女。

加藤　安弘(18)滋賀県

最後のだけはお金じゃ買えないなあ……。あ，でもバイト先で出会いがあるかも。

◆先日，NHK教育テレビの「海外ドキュメンタリー」のCGA特集を見ていて，そのCGA制作者の根気に感心してしまった。すごいなあ。CGAの3D技術も上がって，数年後には，ちょっと見では現実の人間と変わらないものもできてしまいそうです。そうなると，OVA「メガゾーン23」の時祭イヴも夢でなくなるかもしれませんね。

小林　裕昭(22)東京都

思いどおりのものが作れて，それを自由自在に動かせるって，カッコいいですね！　ほぉら，あなたもやりたくなったでしょ。

◆夢の中でアセンブラのプログラミングをすることほど，気分の悪いものはほかにあるまい。

河野　太郎(19)東京都

夢の中でバグを発見する，という体験をした人もいますよ。君ィ，修業が足りんよ。なんてね。

◆私は何かを揃えることはすごく好きなんだが，実際に役立てているのかというとそうでもない。この1年半あまりでゲームソフトが約40本。たいてい，やり尽くさないまま放り出し，エンディングを見てないのもいくつかある。プログラミングをしようと買った「THE福袋V2.0」も，そのまま。何もやってない。今度はOh!Xのうしろにあった X68K Programming Seriesを買おうかなと思ってる。恐ろしい。　吉本　康孝(20)福岡県

そうですか。あなたは「コレクター」なのですね。私？　先立つものが……。くすん。

◆数学のテストで「おいしいカレーの作り方」を書いて点をもらったという話を聞いた。私はこんど，宮沢りえ好きの化学の先生のテストは，テスト用紙を宮沢りえのコラージュにしようかと思っている。いや～大学の先生って話せるな。高校のときなんか，歴史のテストで「～の時代の背景を書け」という問題に，田んぼや山を描いたらペケだったもんな(当然といえば当然だが)。

鈴木　武虎(18)愛知県

あのう……テスト用紙にコラージュって，時間内にできるのでしょうか？　なんか，化学の勉強をしたほうが手っとり早いような気もするのですが……。まあ，とにかく健闘を祈ります！　めでたく単位をもらったあかつきには，ぜひとも，その「作品」をOh!Xにも投稿してね。

◆そろそろ修論をまとめないとスキーに行けない。でも，バブルがはじけてネタがない。

五十嵐　豊(25)千葉県

バブルがはじけるとなくなってしまう？　いったい，テーマはなんなの……？？？

◆とにかく，いそがしかった！　何がいそがしいかって，9/24～30(土・日を除く)にうちの会社で所属対抗バレーボール大会というのがありまして，この間うちの課のチームは勝ち続け，その結果，試合後は飲み会。水曜を除いて飲みっぱなし！　9/30に負けたのでやっと飲み会地獄から解放されたかと思ったら，10/3，4と課内旅行。またもや飲み会の嵐！　そしたら10/6には課内送別会でまたまた飲み会！　ほんとにつかれた。

本田　英雄(23)埼玉県

飲み会地獄！　う，うらやましい。ひと声かけてくだされば，よろこんで「身代わり」になってあげたのに……。

◆音楽！　私には悪夢のような思い出があります。いまから6年ほど前，私の弟はヘビメタ(「アンセム」とかいうグループ)のボーカルで，渋谷公会堂での公演にムリヤリ呼ばれたことがあります。周囲は16～17歳のヘビメタヤング！　みんな，イスの上に総立ちして，歓声の嵐！　狂気の集団！　……静かなクラシック・リスナーの私には(今なお)信じられぬ異世界でした。何万人もの群衆の中の異邦人と化した私は，「群衆の中の孤独」を思いきり味わいました。怖かった。

坂本　慎太郎(31)北海道

音楽って，感性に合わないものはただの騒音っていいますからねえ。それにしても，ヘビメタもクラシックも聴く私は，幸せ者？　それともただの節操なし？

◆バブル崩壊のため，世の中が不景気となり，入社半年にして自宅待機となってしまいました。友達も10月いっぱいでクビになってしまうそうです。社内に嫌いな人が多く，会社なんて辞めてしまいたいと思ってはいたのですが，自宅待機となってしまうと，早く会社に戻りたい，ただそれだけを考えています。結局わがままな人間なのです，私は。

また，最近他人の嫌なところがよくみえるようになりました。私がこのように感じているのだから，私も他人様から同様な感情を抱かれていると思います。完璧な人間というのは無理ですので，そのなかにあって，個性は消さずに日

▲尾形　雅治　広島県
5周年へのカード，どうもありがとうございます。
Oh!Xも，りぼんをつけてもらって，はっぴー！
長いようで短い5年間。これからもよろしくね。

▲横井　賢一　富山県
きゃあ，かわいい！　いろいろな服を着たエトワール・プリンセスを描いてみるのも楽しいかもしれませんね。彼女もきっと喜んでくれるでしょう。

頃の態度を改めてみようと思います。

前田 賢一(20)静岡県

仕事や会社って嫌なこともありますが，いざ，なくなっちゃうと，やっぱりつらいでしょうね……(しみじみ)。でも，ここでどうするかが，これからの分かれ目かも。いろいろなことを再認識するための時間を与えられたのかもしれません。「充電期間」だと思って，くじけずにがんばってくださいね。そのうち，うれしいご報告がくることをお待ちしていますからね！

◆10月号171ページの「CD-ROM Drive for X68000売り上げ倍増計画」は，本当は友人「伊藤彰」君の発案なんです。すみませんが，そういうことですのでつつしんでおわび申し上げます。次回こそ自分のネタで載るぞ！

髙橋 大介(21)愛知県

おやおや，そうだったのですか。それならば，「三人寄れば文殊の知恵」とか「矢も三本集まれば折れない」とかいうことですし，お友達もうひとりを加えて，さらなる最強(？)の倍増計画，お願いしますよ。

◆マリオペイントで音楽に目ざめた私はMUSIC PRO-68Kを買いに走ったが，ないのでMUSIC STUDIOを買ってしまった。だけど，これはMIDIがないと動かないのだ。まだまだ目ざめはこないようだ。

新井 一成(22)埼玉県

目ざめはもう始まってるじゃないですか。今度は，音楽関連の投稿もねっ。

◀小川 裕美 山口県
この夢みるような瞳で魔法をかけられたら，もう誰も逆らえない！ ということで，「ゆーわく」に負けてしまった担当者です……。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できないこともあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★「西夢(とりむ)」では，新規会員を募集します。月に1回，X68000用のディスクマガジン発行を主な活動としています。会員の方にプログラム＆データを投稿してもらい，編集本部がまとめるという，電脳倶楽部のようなサークルです。興味のある方は500円分の郵便小為替を同封のうえ，下記の住所に連絡してください。〒514 三重県津市渋見町630-3 西 敬史方「西夢」

★このたび，「CUREC」では12月発行予定の「DISK CUREC MAGAZINE Vol.7」より内容を一層充実させ，活動を本格化することになりました。そこで第2次会員を募集します。毎月2枚のディスク会報は，文章やプログラムなど盛りだくさんの内容です。「誰もが気軽に参加できる活動」を前提としていますので，初心者からプロの方までX68000を持っていれば年齢，性別，経験などいっさい問いません。「CUREC」の活動に興味を持たれた方，ぜひ入会したいと思われた方は，下記の住所まで120円切手を2枚同封してお送りください。折り返し入会申込書と活動内容を説明したサンプルディスクをお送りします。〒488 愛知県尾張旭市新居町今池2911-2水野方 J.X.U.C.CUREC「第2次会員募集」係

売ります

★アイテックのSCSIハードディスク「TX-130」(130Mバイト)を送料込み50,000円，X68000 XVI用2Mバイト増設メモリ「CZ-6BE2A」「CZ-6BE2B」×2を，それぞれ送料込み25,000円で売ります。「TX-130」は箱，付属品あり，「CZ-6BE2A」「CZ-6BE2B」は本体のみです。まずは官製ハガキで連絡してください。〒097 北海道稚内市緑1-10-23 小川アパート2号室 山崎紀明(22)

★X68000用SCSIボード「CZ-6BS1」を15,000円前後，IOデータ製X68000用2Mバイト増設メモリ「PIO-6BE2」を30,000円前後で売ります。すべて箱，説明書あり。希望価格を書いて往復ハガキで連絡してください。〒520-30 滋賀県栗太郡栗東町出庭630-1 井村 英二(21)

★X68000用カラーイメージユニット「CZ-6UTI-BK」を34,800円，14型ディスプレイ「CZ-603D-BK」を25,000円，テレビチューナー「CZ-6TU-BK」を16,000円で売ります。すべて箱，マニュアル，付属品つきです。連絡は往復ハガキでお願いします。〒380 長野県長野市大字稲葉2638 倉石 知徳(29)

★Roland「CM-64」＋カード1枚を65,000円(送料込み)で売ります。新品同様，保証書あり，ただし箱なし。連絡は電話番号を明記のうえ官製ハガキでお願いします。〒816 福岡県春日市小倉213-1 春日サンハイツ9-504 井上 圭次郎(19)

買います

★MZ-1500/2500用FDD「MZ-1F07」を20,000円以下，MZ-2500用増設RAM「MZ-1R26」を5,000円程度，MZ-2500用マウス「MZ-1X10」または同等品を2,000円程度で買います。連絡は官製ハガキ，封書，または往復ハガキでお願いします。〒050 北海道室蘭市水元町25-21 ハイツラポートC206号 羽生 知浩(20)

★ハンディスキャナ「HGS-68」(完動，付属品つき)を送料込み20,000円で買います。連絡は往復ハガキでお願いします。〒889-21 宮崎県宮崎市大字加江田537 長峰 孝文(27)

★X68000用拡張I/Oボックス「CZ-6EB1-BK」(黒)を60,000円前後で買います。多少の傷はかまいません。また，ファミリーコンピュータ用の光線銃を50,000円くらいで買います。希望の価格を書いて往復ハガキで連絡してください。〒762 香川県坂出市白金町3-8-34 佐藤 崇司(18)

★X68000用数値演算プロセッサボード「CZ-6BP1」を40,000円程度，X68000用拡張I/Oボックス「CZ-6EB1-BK」を40,000円程度，X68000用SCSIボード「CZ-6BS1」を15,000円程度で買います。なるべく付属品，マニュアルつきでお願いします。連絡は品物の状態を書いて，往復ハガキでお願いします。〒491 愛知県一宮市丹陽町三ツ井1667-2 井上 宏一(19)

バックナンバー

★Oh!X1991年10月号，1992年3月号を各1,500円(送料込み)で買います。切り抜きのあるものは不可。連絡は往復ハガキで。〒180 東京都武蔵野市西久保2-27-5 木場 健郎(14)

★Oh!X1991年6月号を2,000円(送料込み)で買います。「PC-9801マウスを使う」の記事が残っていればかまいません。まずは官製ハガキで連絡をください。〒194-02 東京都町田市小山田桜台2-5-25-201 長野 充宏(18)

★Oh!X1990年6月号を1,000円(送料込み)で買います。付録ディスクはなくてもかまいませんが，切り抜きのあるものは不可。連絡は往復ハガキで付録ディスクの有無を明記のうえお願いします。〒631 奈良県奈良市西大寺秋篠早月町317 アーバンスクエア103 服部 亘(19)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々の意見を紹介しています。今月は10月号の内容に関するレポートです。

●現在のDTMは，ステップ入力ができてコントロールチェンジやら，複雑なコマンドを使いこなせる人たちだけのためにあるような気がします。音楽ってもっと身近で楽しいはずだと思いませんか。高級なソフトが出るより，譜面入力のできる誰にでも楽しめるソフトがもっと充実すればいいと思っています。あとCD-ROMの紹介がありましたが，私がCD-ROMに望むもの，それは絵本の世界です。現在，子供向けのビデオが多数出回っていますが，出ている声優さんはみな子供に馴染みのない人ばかりです。そこでアドベンチャーとまでいいませんが，子供が自分で話を進めていけるような馴染みのある声と，自由度の高い絵本が出てほしいですね。子供って大人の真似をしたがるのです。だから子供向けのちょっと小さめのマウスをつないで，大人と同じことをさせると喜びますよ。

野原　志貴乃(30)　X68000 ACE-HD　埼玉県

●特集ではローランドのMIDI音源や，MUSIC PRO-68K［MIDI］などの使い方が具体的に解説されていて，現在の標準システムを活用するための基本的な説明，という感じがしました。私は10月号の特集に，現状から近い将来にわたる音楽環境全体の概観を期待していたので，ちょっともの足りない感じです。GM,GSの方向性，Z-MUSICの今後，データの共有などそういったものを検討して，今後求められる音楽環境を提案するような記事もほしかったと思います。特集で印象に残った記事は，「Z-MUSICでバビンチョ」でした。これは，高速化というZ-MUSICのもうひとつの側面を見るようで面白かったです。私がZ-MUSICに求めるのは，簡潔でわかりやすいソースと豊かな表現力なので，共通コマンドはもちろんPCM8モードも使うのが当たり前，と思っている部分があります。しかし，アクションゲームのBGM用として使う場合，すさまじい努力が要求されるのですね。私は今回の記事を読んで，Z-MUSICもひたすら速いアクションゲーム用とひたすら高機能，簡潔さを追及したDTM用の２つに分かれるべきだと思いました。そして，両者の橋渡しをするコンバータやプリプロセッサなどがあれば完璧でしょう。

宍戸　輝光(18)　X68000 PRO,MSX2　東京都

●最初に「SX-WINDOW開発キット」の話を聞いたときには，やった！　SX-WINDOW用のBASICみたいな言語が出るのか！　と思いましたが，C言語やアセンブラで開発するためのものと聞いて非常にがっかりしました。SX-WINDOWには私もチャレンジしてみたのですが，あえなく挫折しました。それは，覚える約束ごとが多すぎるからです。実際に私たちがウィンドウでやりたい作業というのは，入力されたものを計算し，絵やグラフを表示する，という最低限のものさえあれば，たいてい実現可能だと思うのです。現在のようになにかやるたびにスケルトンがこうで，というのではちょっと手を出せません。そういう意味でも以前Oh!Xでちらりと話が出た，SX-BASICというようなものが出てこないかと期待しています。

湯沢　聡(29)　X68000，X1turboIII，MZ-2531，PC-1360K，MSX/MSX2，PC-6601　埼玉県

●CD-ROMというのは，そのメディアの性格上ソフトがないとどうにもなりません。プログラムに関しては，ソフトハウスがX68000用を出してくれるのを待つしかないため，現在の段階では紹介記事にあるようにデータのみを利用することになります。そこで問題なのが，面白いまたは役に立つデータがどこにあるのか，そしてどのようなものがあるのか，ということです。もちろんX68000でそれらのデータを読めるのか，もし読めるのならばどのようにしたらいいのかということです。一般の人にとって，まだそういったことを知る手がかりが少なすぎます。そういう情報をまとめたもの（CD-ROMで供給するとか）があるといいのですが。

矢野　啓介(19)　X68000 XVI,MZ-2500　北海道

ごめんなさいのコーナー

9月号　FPP.MACの作成

P.90　リスト１のFPP.MACが正常にアセンブルできませんでした。リスト１の99～108行までの先頭にある‘＊’を削除して，同じリスト１の77～97行を削除してください。

なお，HAS.X ver2.34以降ではマクロ命令の仕様の違いから正常にアセンブルできません。シャープ純正のAS.X ver.2.0かHAS.X ver2.43以前のバージョンを使用してアセンブルしてください。

10月号　ペンギン情報コーナー

P.162　Sapporo Multimedia&CG'92実行委員会事務局の問い合わせ先がFAX番号になっていました。正しくは，☎011(210)9221です。ご迷惑をかけた関係者の皆様に深くお詫びいたします。申し訳ありませんでした。

11月号　SAVESC.SYS

P.32　本文中の説明にあった起動キー一覧表が抜けていました。以下に一覧表を掲載します。

表１　起動キー一覧表

通常	カーソルキー
パワーオフベクタが書き換えられている場合	ROLL UP,ROLL DOWN,UNDO

11月号　EDIT

P.46　１行に75文字以上のテキストをエディットしようとすると画面がスクロールしてしまい，正常にエディットできないことが判明しました。まず，3341$_H$番地からの５バイトを，

```
3341 3A → CD     ＊CALL $3ABB
3342 5C → BB
3343 1F → 3A
3344 67 → C6     ＊ADD A,5
3345 7D → 05
```

以上のように書き換え，3ABB$_H$番地以降の６バイトに以下のよう追加してください。

```
3ABB  3A 5C 1F  ＊LD A,($1F5C)
3ABE  67 7D     ＊LD H,A : LD A,L
3AC0  C9        ＊RET
```

バグに関するお問い合わせは
☎03(5488)1311(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

Oh!X流3分間プログラミング！だったらいいね

▶Oh!X5周年記念特別企画・ショートプロ大集合，いかがでしたか。最近，(で)のショートプロぱーてぃでしか紹介のなかった，手軽に遊べて役に立つショートプログラムを，10人のライターたちに制作してもらいました。懐かしい作品，ちょっと便利なツール，それぞれのライターの個性があふれる楽しいショートプログラムです。プログラム制作に当たっては，ショートプログラムということで長さの制限もあり，ぞれのライターたちは結構苦しんだようです。思い切って仕様を切り詰めたり，素直にあきらめ手を抜いたり，ショートプログラムで収まるようなテーマを選んだり。それでもひとつの完成された作品として，読者の皆さんは評価してみてください。
▶また，このような小さなプログラムを組み合わせることで，次のステップに進むことができることを忘れてはいけません。特別企画で紹介したショートプログラムも，大きなプログラムになる可能性を秘めています。これからプログラミングを始めようとする人は，目標を高く，作業は地道にステップアップさせることを念頭において，がんばってみてください。時間はかかりますが，作品を完成させたノウハウは，必ず次の作品に生かされますから。
▶予告にあるとおり，来月はハードウェア関係の特集です。今回は比較的おとなしいものになりそうだ，と担当者はいってましたがどうなることやら。そして，なにやら怪しい計画の発表をどさくさ紛れに行ってしまおう，という予定もあります。
　また，今月のペンギン情報コーナーで紹介されていた，X68000用サブボード「POLYPHON」を詳しくレポートします。すでに「PCM8ボード（俗称）」としてご存じの方もいると思いますが，具体的にどのようなものなのか，こちらもお楽しみに。
▶「よいこのSX-WINDOW」は，著者多忙のためお休みさせていただきます。楽しみにしていただいた方には本当に申し訳ありません。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先
〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「テーマ名」係

SHIFT・BREAK

▶最近学校をよく休む。とはいえ授業があるわけではなく卒業研究なので，成果を上げれば出席などは関係ないのである。おかげで，休むなら連続で休むよりも，来たり休んだりと分散させるほうが目立たないことを発見してしまった。うまくやって週の半分を休みにできたらいいのだが，そうは問屋がおろさないようだ。世間は厳しいなぁ，うんうん。(八)
▶遅ればせながらYMOのTECHNO BIBLEを買った。やっぱりよい。聴きながら思わず涙した。それと最近クライズラー＆カンパニーというのも好きだ。いまの仕事中のBGMはたいていこれ。あとチェックしている人も多いだろうが，フジの朝番ウゴウゴルーガも脳細胞を心地よく刺激してくれる。やっぱり冬になると調子がいい。ふふふ。　冬生まれの（哲）
▶新聞を読んでいたら「天皇皇后両陛下，見たこともない生物を見学」という記事があった。今年8月に発見されたそうで，新華社通信いわく「動物と植物の両方の特徴をもつ生物」だそうだが，すごいのが「付近の住民が食べてみたらおいしかった」という解説。さすが中国。未知の生物を食うなよ。世界に誇る中華料理の秘密を垣間見た気がしたな。(浦)
▶「リーサルウェポン3」を見てきた。1，2，3と回を増すごとに全体の雰囲気が明るくなっているこの映画。パート4はどうなってしまうんだろう。「美女と野獣」も見てきた。たわいのない話なのに結構泣けた。CGとの合成がウリだったらしいが合成がミエミエでいまいち。やはりディズニーは全部手書きでやってほしいね。(善)
▶日曜は暇でしょ，なんていわれることがあるが，兼業ライターである僕にはそんなことはない。でも寸暇を惜しんで原稿を書いているかといえば，そんなこともない。ぼんやりしてたりゲームしてたり，気合を入れて仕事するでもなし，遊びにいくでもなし。中島らもの本に書いていたっけ「僕にはヒマなんてない，あるのはサボリだ」ああ妙に納得。(A.T.)
▶だからOh!Xって好きなんだけど，奇特な読者の方が女の子のPICファイルを送ってくれたので，来月，使うかもしれません。ディスク2枚組の大作です。やはり，いってはみるものです。あ，ヌードじゃないから安心するように。ちなみに，私がファンクラブの初代16階調です。大丈夫です。愛してます。好きだから，それだけです（笑）。(K)
▶発売初日で完売できなくて新聞ネタにもなったドラクエV。僕も初日に手に入れてプレイした。感想はゲームバランスが悪いということ。一見，重要そうで不要なアイテムが多い。最後の敵は印象が薄いし弱すぎる。ドラクエらしからぬ裏面の存在も疑問。でも，文句たらたらながら結構楽しんでプレイできたのは，さすがドラクエというべきかな。(KO)
▶地下鉄銀座線の渋谷駅ホームに，「ゲリラ注意」と朱文字で大書してあった。過去になにか事件があったのかなあ。どういうふうに注意するんだろう？いったい，どんなカッコしてるのかな，見た途端ソレとわかるのかな。やっぱり走って逃げたほうがいいのかなあ。いろいろ考えていたら，あやうく乗り越しちゃうとこだった。まったく，もう。(ふ)
▶うまく進まなくなると投げる。しばらく寄り道をしてから拾う。また，うまくいかなくなると投げる。そして，また拾う。僕とプログラムはそんなつきあい。プログラミングの結果として求めるのは，コンピュータを使って面白いことをやりたい，それだけ。まだ，思うような結果は得られていないが，いつかそのうちなんとかなるだろう。(J)
▶採るべき道は決まった。486機に張り倒されそうにもなったが目的のソフトがない。Macintoshも購入してみたがほしいソフトは高すぎる。そこへAMIGA4000の登場。安くなったAMIGA3000の直輸入もいいと思うんだけど「AMIGA4000を買ったほうがいい」という熱い要望に負けそうな雰囲気。ボーナスが全部飛んでしまいそう。が，これも定めか。(A)
▶夏が終わったばかりだと思っていたら，もう話題は冬である。冬はせらむん一色に染まりそうな気配とか。アニパロ系，男性向け創作，少女系をはじめ，なんと旧翼トルーパー勢力が移行しつつあるらしい。あの超絶的な人波には絶対に無縁だと思って安心していたのに……。しかし，一度も見たことない番組なのに妙に詳しい自分が怖い。(U)
▶おかげさまで，誌名がOh!Xになって5周年。かつては読者や関係者だった人からも「変わらないねえ」と感心される。でも本当に大切なことはなにも変わらないものだと思う。それに，パソコンの世界は日進月歩なんて誤解されるけど，「幕張に，21世紀あらわる」とかで，行ってみたらPC-9821というのが出ていたなんてね。そういうものでしょ。(T)

microOdyssey

その昔，私は「文学少女」だった。

といっても，べつに「物静かで繊細な美少女」だったわけではない。単に「活字中毒」の子供であっただけのことだ。そのへんにある，文字が書いてあるものはなんでも読んでしまう。内容をすべて覚えるわけではない。とにかく，読む。それは単なる趣味だったので「読んで」さえいれば私は満足だったのだ。

ところが最近，ちょっと困っている。2年ほど前，ひょんなことから「雑誌の編集者」になってしまったのだ。

さあ，大変。

当然のことながら，本が世の中に出るまでには，数回のチェック作業が行われる。ページのすみからすみまで「なめるように」眺めるのが，編集者の仕事のひとつである。誤字・脱字，印刷のずれから，文章表現にいたるまで，である。特に文章については，文芸誌などとは違って，表現の美しさよりも正確で誤解を生まないことが最優先なので，筆者の原稿に手を入れることもしばしばである。その結果として，「重箱のすみをつつく」性分がしみついてしまうのだ。

先日も，ひとりで某アドベンチャーゲームをやっていたときのこと。

私は，小説を読んだり，映画を観たりするときも，わりと，のめり込んでしまう質である。そのゲームは出来もよくて，結構サクサク進んだので，その時も，気分はすっかり敏腕刑事，であった。そして，殺人事件が起き，私は聞き込みを開始した。

被害者は東京在住で，証言者はその秘書，ここ，殺人現場は北海道である。秘書はいう。

「北海道へは観光に出かけたそうですよ」

あああ……。このたった一言の台詞で，私は一瞬にして殺人現場から，夜中にゲームをしているひとりの部屋に，はじき飛ばされてしまった。ここは北海道なのに，どうしてここで「観光に来ていた」っていってはくれないのか……。

すぐに気をとりなおして，こころは再び北海道に飛んで事件は解決したのだが，まあ，こんなことはたいしたことじゃない。ただ，最近，日常生活すべてがこの調子だから困っちゃうのである。言葉というのは，あからさまな間違いではなくても，いいまわしによってどちらにも受け取れることだってある。本などを読んでいても，そういう部分があるとついひっかかってしまう。それが重要なことならよいが，気をつけていないと枝葉末節のところでも，ぐちぐち考え込んでしまう。まったくもって，本末転倒である。これでは「職業病」だなんて笑っている場合ではない。

だいぶ前のことになるが，雑誌に載っていた詩の一節に，こういうのがあった。

　ことばはうそをつきますか
　ことばがこころをうつさぬならば
　ことばはうそをつきますか

そう，言葉は嘘をつくことだってできるのだ。それはひとつの表現上のテクニックだから，善し悪しをいうわけではないが，重要なことだ。言葉っていうのは私達の意志そのものなのではなく，意志を表す道具にすぎないということを，私はつい忘れてしまうことがある。言葉に踊らされてはいけないのだ，と反省することが多い今日この頃。（ふ）

1993年1月号12月18日(金)発売

特集　ハードウェア工作
謎のXXXXXボードプロジェクト
新製品紹介
POLYPHON/THUNDER WORD
X68k Programming Series
X68000 CARDDRV.X用ゲーム　GOLF
全機種共通システム　EDC-Tの拡張

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートBI	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
	池袋	旭屋書店池袋店	03(3986)0311
	八王子	くまざわ書店八王子本店	0426(25)1201
神奈川	横浜	有隣堂横浜駅西口店	045(311)6265
	〃	有隣堂ルミネ店	045(453)0811
	藤沢	有隣堂藤沢店	0466(26)1411
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
新潟	新潟	紀伊國屋書店新潟店	025(241)5281
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 12月号
■1992年12月1日発行　定価600円(本体583円)
■**発行人**　孫　正義
■**編集人**　橋本五郎
■**発売元**　ソフトバンク株式会社
■**出版事業部**　〒108 東京都港区高輪2-19-13　NS高輪ビル
Oh!X編集部　☎03(5488)1309
出版営業部　☎03(5488)1360 FAX 03(5488)1364
広告営業部　☎03(5488)1365
■**印　刷**　凸版印刷株式会社

©1992 SOFTBANK CORP. 雑誌 02179-12 本誌からの無断転載を禁じます。
落丁・乱丁の場合はお取り替えいたします。

満開の電子ちゃん

作・え 岡村祭

講読方法：定期購読もしくはソフトベンダーTAKERUでお買い求めいただけます。

★定期購読の場合＝購読料６ヶ月分6,000円（送料サービス、消費税込）を、現金書留または郵便振替で下記の宛先へお送り下さい。

現金書留の場合：〒171 東京都豊島区長崎1-28-23 Muse西池袋2F ㈱満開製作所
郵便振替の場合：東京 5－362847 ㈱満開製作所

- ご注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
- 3.5インチディスク版をご希望の方は、「3.5インチ版」とご指定下さい。
- 新規購読の方は「新規」と明記して下さい。なお、特に購読開始号のご指定がない場合は既刊の最新号からお送りいたします。
- 製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします。

★TAKERUでお求めの場合＝１部につき1,200円（消費税込）です。

- 定期購読版と内容が一部異なる場合があります。御了承下さい。
- お問い合わせ先　TEL(03)3554－9282（月～金　午前11時～午後６時）

（なお、定期購読版のバックナンバーについては定期購読の方のみご注文を承ります）

谷口 博一（大阪府）

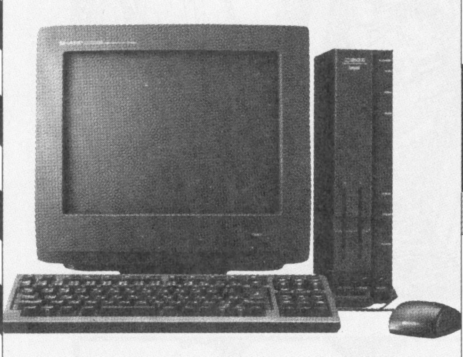
三信七疑で電脳倶楽部の購読を始めた私ですが、いまでは毎日の生活に必要不可欠なものとなってしまいました。

あれは忘れもしない×年前の事、友人宅で初めて電倶を見た時に、『こ、こんなものが許されるのか』と、思わず吠えてしまいました。それ以来、私は電倶の虜となってしまい、もう抜け出すこともできない状態です。一見アブナそうで、実はたいへん危険なものですが、受験と同様、人生の賭が必要です。タケル購読より定期購読の方が、より危険性が増し、エクスタシーを感じることでしょう（大ウソ）。

マイコンショップ川口

☎048-225-1718

（消費税別）



New X68000
COMPACT XVI
~~¥298,000~~

CZ-674C-H	¥298,000
CZ-608D-H	¥ 94,800
AV-090-SC	¥168,000

超特価 TEL下さい

CZ-634C-TN	㊤368,000
CZ-644C-TN	㊤518,000

ソフト各種超特価ご奉仕中

CZ-219SS OS-9/X68000	定価¥29,800
CZ-213MS MUSIC PRO68K	定価¥18,800
CZ-214MS SOUND PRO68K	定価¥15,800
CZ-215MS Sampling PRO68K	定価¥17,800
CZ-220BS DATA PRO68K	定価¥58,000
CZ-224LS The 福袋 Ver2.0	定価¥ 9,980
CZ-225BS Multiword	定価¥32,000
CZ-251BS Hyper word	定価¥39,800

中古売買価格表

品　名	買取り価格	売　価
CZ-633C	160,000より	180,000より
CZ-644C	210,000より	230,000より
CZ-613C	105,000より	125,000より
CZ-603C	75,000より	95,000より
CZ-612C	85,000より	98,000より
CZ-602C	65,000より	85,000より
CZ-653C	75,000より	95,000より
CZ-663C	95,000より	115,000より
CZ-662C	75,000より	98,000より
CZ-652C	55,000より	75,000より
CZ-611C	70,000より	89,000より
CZ-601C	45,000より	65,000より
CZ-612D	35,000より	45,000より
CZ-602D	30,000より	39,800より
CZ-603D	20,000より	29,800より
CZ-604D	25,000より	34,800より
CZ-605D	45,000より	55,000より

プリンター

CZ-6VT1	特価¥ 47,700
CZ-8PG1	特価¥ 86,800
CZ-8PG2	特価¥106,900
CZ-8PK10	特価¥ 66,800
CZ-8NS1	特価¥141,000
CZ-6BC1	特価¥
CZ-6BG1	特価¥
CZ-6BP1	特価¥
CZ-6BP2	特価¥ 34,400

ラムボード

CZ-6BE2A	定価¥59,800	特価¥ 44,900
CZ-6BE2B	定価¥54,800	特価¥ 41,100
CZ-6BE2D	定価¥54,800	特価¥ 41,100
CZ-6BE1B	定価¥28,000	特価¥ 21,000
CZ-6BE2	定価¥79,800	特価¥
CZ-6BE4C	定価¥98,000	特価¥
PIO-6BE1-A	定価¥25,000	特価¥
PIO-6BE2-2M	定価¥50,000	特価¥
PIO-6BE4-4M	定価¥88,000	特価¥
SH-6BE1-1M	定価¥25,000	特価¥

ファイル

CZ-6MO1	定価¥450,000	特価¥
CZ-64H	定価¥120,000	特価¥
CZ-68H	定価¥160,000	特価¥

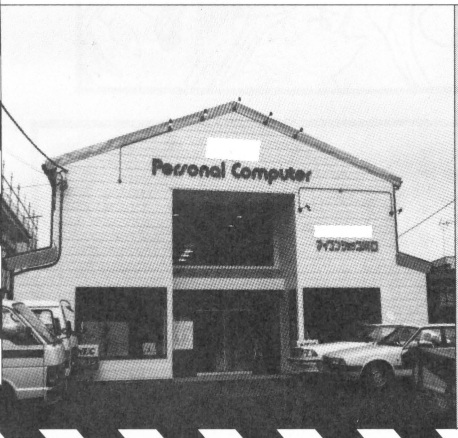
その他機種

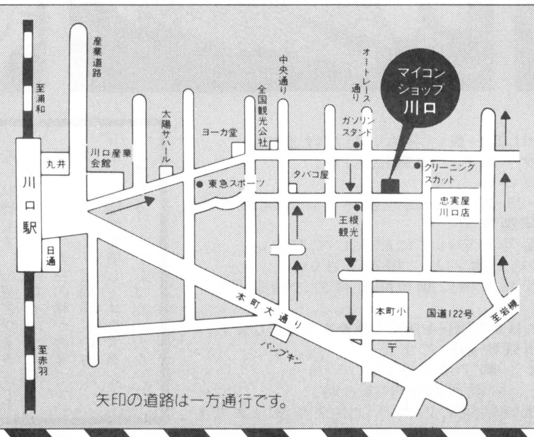
CZ-8NS1	カラーイメージスキャナ	定価¥188,000	特価¥
JX-220X	カラーイメージスキャナ	定価¥168,000	特価¥
CZ-6BN1	スキャナ用パラレルボード	定価¥ 29,800	特価¥
CZ-6VT1	カラーイメージユニット	定価¥ 69,800	特価¥
CZ-6BV1	ビデオボード	定価¥ 21,000	特価¥
CZ-8TM2	モデムユニット	定価¥ 49,800	特価¥
CZ-8NJ2	インテリジェントコントローラ	定価¥ 23,800	特価¥
CZ-8NM3	マウス・トラックボール	定価¥ 9,800	特価¥
CZ-8NT1	トラックボール	定価¥ 6,888	特価¥
CZ-8NJ1	ジョイカード	定価¥ 1,700	特価¥
CZ-6BC1	FAXボード	定価¥ 79,800	特価¥
CZ-6BM1A	MIDIボード	定価¥ 26,800	特価¥
CZ-6BP1	数値演算プロセッサ	定価¥ 79,800	特価¥
CZ-6BP2	数値演算プロセッサ	定価¥ 45,800	特価¥
CZ-6TU-BK-GY	RGBシステムチューナー	定価¥ 33,100	特価¥

★クレジット回数1～60回まで設定自由

回　数	1	3	6	12	15	20	24	36	42	48	54	60
金利(%)	2.5	2.9	3.9	5.4	8.4	10.9	11.4	15.9	19.9	20.9	25.9	26.9

中古品も取扱っております。





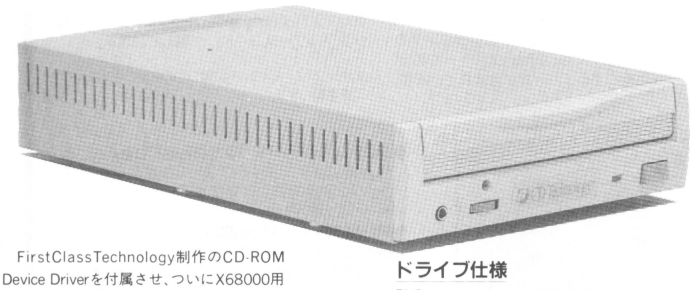
矢印の道路は一方通行です。

通信販売をご利用の方

—全国通販—

通信販売をご利用の方は、売値の変動がありますので在庫、値段をあらかじめ確認のうえ電話で、商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。

絶賛発売中！
CD-ROM Drive for X68000
マルチメディアへの誘い



FirstClassTechnology制作のCD-ROM Device Driverを付属させ、ついにX68000用CD-ROM Driveの登場です。本製品を使用することにより、MS-DOSやPC-9801シリーズ、FM-TOWNSなどで採用されている、ISO9660規格のCDをHuman68K SX-WINDOWで直接扱えるようになります。

また、将来の拡張にも柔軟に対応できるSCSIインターフェースによる接続を採用。ディジーチェーンによって既存のSCSIハードディスクとの同時使用も可能です。

標準価格¥118,000-

ドライブ仕様

型番	KGU-XCD
使用ドライブ	東芝　XM-3301
平均アクセスタイム	325mSEC
インターフェース	SCSI
キャッシュメモリー	64KB
オーデオ出力	RCA-Phono端子×2 ステレオヘッドホン端子
電源	専用ACアダプター
外形寸法	150×228×50（電源部含まず）

※SCSIケーブル・ターミネーターは別売になります

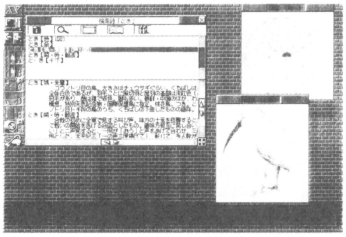
付属サポートソフト

ISO9660準拠デバイスドライバ
MusicPlayer for SX-Window
Macintosh™用ファイルビューア for SX-Window

KGU-XCD対応　X68000 CD-ROM第一弾！「フリーウェア集」 **Free Soft Ware Selection-CD68K** 近日発売

X68000 Pro SHOP
BASIC HOUSE
KEISOKUGIKEN Corp.
TEL 0286-22-9811　FAX 0286-25-3970

CD-ROM広辞苑 検索&閲覧
SX-広辞苑



富士通・NECより発売されている12cmCD-ROM広辞苑を検索/参照し、X68000上で使用できるようにする専用検索ソフトです。通常の検索に加えて、文章中語検索、条件検索などの強力な検索機能が利用でき、広辞苑内に収録されている図版の表示、音声の再生も可能です。SX-WINDOW上で動作するので、SX-WINDOWの特徴である疑似マルチタスク機能やカット＆ペースト等の機能が利用でき、エディタXなどで文章編集中に検索/参照することが簡単にできます。また将来ワープロなどのSX-WINDOW対応ソフトが発売された場合にも本ソフトの使用によって広辞苑の有効利用が可能になります。同時に複数起動ができるため、いくつかの情報を同時に参照しながら作業を進めることも可能です。

標準価格　検索ソフトのみ　¥19,800-
　　　　　CD広辞苑セット　¥45,000-

※本アプリケーションの実行にはメインメモリ2Mバイト以上実装している必要があります。

通　販　大　特　価

X68000 Compact XVI

CZ-606D/CZ-6FD5 SET

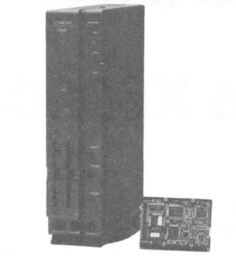
定価合計￥477,600
特価￥338,000

X68000 Compact XVI

CZ-606D/CZ-8PC5-BK SET

定価合計￥474,600
特価￥358,000

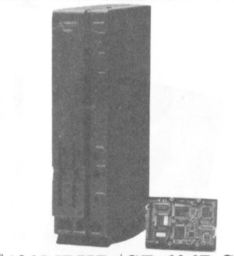
X68000 Compact HD80



2.5″80MBHD/CZ-606D SET

定価合計￥545,800
特価￥399,800

X68000 Compact HD120

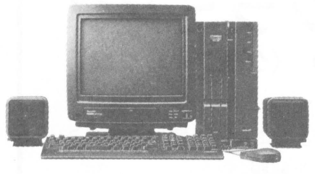


2.5″120MBHD/CZ-606D SET

定価合計￥575,800
特価￥428,000

※Compact HD80/120は、ベーシックハウスのオリジナル商品です。シャープ㈱製HDキッドとは別商品ですので御注意下さい。

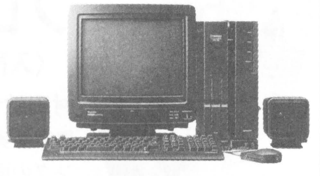
X68000 XVI HD100



CZ-606D/Quantum100SET

定価合計￥565,800
特価￥398,000

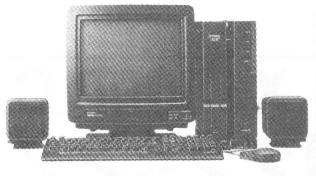
X68000 XVI HD120



CZ-606D/Quantum120SET

定価合計￥575,800
特価￥408,000

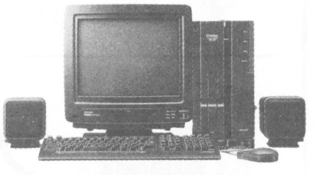
X68000 XVI HD240



CZ-606D/Quantum240SET

定価合計￥625,800
特価￥438,000

X68000 XVI HD425



CZ-606D/Quantum425SET

定価合計￥745,800
特価￥528,000

低金利クレジット　通信販売送料 全国一律￥1,000 長期クレジット可能
※表示価格に消費税は含まれておりません

株式会社 計測技研　マイコンショップ BASIC HOUSE
本社／ショールーム／通販部
〒321 栃木県宇都宮市竹林町503-1
TEL 0286-22-9811　FAX 0286-25-3970

OS-9 Ver.2.4 対応パッケージリリース

microware

OS-9/X68000 C＆Professional Pack. V3.2

C＆プロフェッショナル・パッケージは、OS-9/X68000上で動作するマイクロウェア・Cコンパイラとユーティリティ・ソフトのパッケージです。

◆マイクロウェア・Cコンパイラの特徴

他OSの、K&R準拠、ANSI準拠のアプリケーション、あるいはUNIX上のアプリケーションは、特に修正することなく容易に移植できます。

このコンパイラはCPUのインストラクションに最適化されており、生成されるオブジェクトが、最小、最速になるように複数レベルに渡るオプティマイズを実行しています。

◆拡張機能

1. シンボリック・デバッグ
2. 強力なエラー診断機能
3. 高速なコンパイル・スピード
4. 豊富なオプション
5. アセンブリ言語とのインタフェース
6. OS-9/X68000用拡張ライブラリ

※バージョンアップサービスを予定しておりますので、お早めにユーザ登録をお済ませ下さい。

◆付属ユーティリティ・ソフト

●SrcDbg(ソース・レベル・デバッガ)

SrcDbgは、C言語で書かれたプログラムのテストやデバッグをソース・レベルで行うユーティリティです。

●μMACS(マイクロマックス)

μMACSは、UNIX上で広く利用されているスクリーンエディタ"EMACS"のOS-9版サブセットです。

◆パッケージ内容

マイクロウェア・Cコンパイラ
標準ライブラリ
OS-9/X68000専用ライブラリ
ヘッダ・ファイル
OS-9/X68000専用ヘッダ・ファイル
アセンブラ
リンカ
ユーザステート・シンボリック・デバッガ
ソース・レベル・デバッガ
漢字フル・スクリーン・エディタ

◆付属マニュアル

Cコンパイラ・ユーザーズ・マニュアル
アセンブラ・リンカ・デバッガ・ユーザーズ・マニュアル
ソース・レベル・デバッガ・ユーザーズ・マニュアル
μMACSユーザーズ・マニュアル
OS-9/X68000専用ライブラリ・マニュアル1,2

3.5"2HD 5"2HD 2枚組
定価¥80,000

OS-9/X68000 テクニカル・デベロップメント・キット Technical Development Kit V2.4

OS-9/X68000テクニカル・デベロップメント・キットには、OS-9上でのプログラミングのためのマニュアルとシステム・ステートでのデバッグを可能とするデバッガが含まれています。

また、デバイス・ドライバ作成のために、※各種サンプルソースコードが付属しています。

※サンプルソースコードに関してのお問い合わせはご遠慮願います。

◆パッケージ内容

マニュアル
　システムコール
　テクニカル
　I/Oテクニカル
　システムステート・デバッガ・ユーザーズ
　ROMデバッガ・ユーザーズ
ソフトウェア
　SysDbg
　RomBUG
　※各種サンプル・ソースコード

◆システムステート・デバッガ(SysDbg)

SysDbgは、OS-9システムの拡張など、I/Oドライバの開発を強力に支援するシンボリック・ステート・デバッガです。

◆ROMデバッガ(RomBUG)

RomBUGは、OS-9とは独立したデバッガです。起動時に必要なコンソールやディスクなどのデバイス・ドライバをデバッグすることができます。

供給メディア
3.5"2HD 5"2HD
定価 ¥38,000円

＊会社名・製品名は、各社の商標または登録商標です。
※製品の内容等は予告なく変更されることがあります。

OS-9/X68000はシャープ㈱から販売しています。

マイクロウェア・システムズ株式会社
〒101 東京都千代田区外神田2-17-3 代表(03)3257-9000 Fax(03)3257-9200

X68000ユーザー必読書!!

Inside X68000

桑野雅彦 著
画面制御関連はもちろん、LSIについても詳説。GCCによるサンプルプログラム付。
定価6,800円

GNUツールボックス

中森章 著
X68000上でのCプログラム作成について、初歩からわかりやすく解説。
定価2,600円

X68000 C プログラミング

吉野智興・村上敬一郎 著
GNUをX68000に移植するためのノウハウについて解説。定価2,200円

SX-WINDOW プログラミング

吉沢正敏 著
内部解説にもとづいたプログラミングの実例を解説。定価4,500円

追補版一定価4,200円
[Ver.1.10対応/ディスク付]

X68000マシン語プログラミング 〈入門編〉〈グラフィック編〉

村田敏幸 著
豊富な実例とともにマシン語プログラミングの面白さを解説。
入門編一定価2,800円
グラフィック編一定価3,600円[ディスク付]

定価は税込みです。

ソフトバンク株式会社/出版事業部

AIBIT ポケコン各種大特価

アイビット電子株式会社

12月15日迄

アセンブラ・C言語・BASIC・CASLの四言語搭載
業界初、16ビットCPU(インテル8086系)搭載
●64Kバイト RAM標準装備
(最大96Kバイトまで拡張可能)
FX-890P
AD-4177J付
定価 ¥37,800➡
¥28,000
CASIO

●C言語・BASIC・CASLの三言語搭載
FX-870P
定価 ¥28,000➡
特価¥21,000
CASIO

●64Kバイト RAM標準装備
(最大128Kバイトまで拡張可能)
●接続ケーブル(CE-140T)で各種パソコンと簡単に接続可能
PC-E550
定価 ¥32,000➡**特価**
SHARP

●32Kバイト RAM標準装備
(最大96Kバイトまで拡張可能)
●接続ケーブル(CE-140T)で各種パソコンと簡単に接続可能
PC-E500
定価 ¥28,800➡
¥19,800
SHARP

●接続ケーブル(CE-T800)で各種パソコンと簡単に接続可能
●32Kバイト RAM
PC-E200
定価 ¥22,000➡
特価¥17,800
SHARP

PC-1360 +CE-140P
定価 ¥72,800➡
特価¥25,000
SHARP

27関数・機能搭載
FX-820P
定価 ¥29,800
特価¥19,800
CASIO

PC-1248DB
定価 ¥11,000➡
特価¥9,800
SHARP

PC-1280
定価 ¥24,800➡
特価¥21,800
SHARP

65関数・機能搭載
FX-795P
定価 ¥19,800➡
特価¥15,800
CASIO

PC-1360K
定価 ¥36,800➡
特価¥32,800
SHARP

PC-1600K
定価 ¥69,800➡
特価¥59,800
SHARP

さわって覚えるBASIC入門機
PB-120
定価 ¥12,800➡
特価¥9,800
CASIO

FX-840PL
定価 ¥34,800➡
特価¥6,000
CASIO

フロッピーディスクボックス
MD-100A
定価 ¥49,800➡
特価¥37,300
CASIO

〈全商品新品完全保証付〉
★シャープ・シャープ周辺機器(拡張機器全機種、プリンター他)・富士通・NEC常時取り扱い。
★シャープ・カシオポケコン全機種取り扱い。PACIFIC・YHP・キャノンも取り扱い。
★学校、企業納入受け賜ります。送料一律¥700。★上記商品価格には、消費税は含まれておりません。
★特価表及び資料をご希望の方は、72円切手を同封の上お送りください。

通信販売のお問い合せ、御注文は
TEL.0426-45-3001(本店) FAX.0426-44-6002
●営業時間/10:00~19:00●電話受付/9:00~22:00迄可●定休日/水曜日
SHARP SUPER EXE SHOP
アイビット電子株式会社 〒192 東京都八王子市北野町560-5

上記の広告商品はすべて店頭販売もしております。
全国通信販売
北海道から沖縄まで
★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。
富士銀行八王子支店 (普)1752505

City Soft X68000 FEP RECALL

より強力な新次元環境へ

FIXER Ver.4.0は構文意味解析処理による高い変換効率を実現。多彩な変換モード(逐次自動変換・一括変換・高速変換・句読点による変換)を装備して使用用途に合った環境を提供。キー操作も標準添付(ASK68K)のFEPと上位互換になっているので、買ったその日から違和感なく使用できます。

その他の機能内容
● ASK標準のFPコール準拠対応。
WP等のアプリケーション上でも使用が可能になります。
● キーカスタマイズ
変換に使用するキーの配置が自由に設定できます。

価格:19,800円

SHARP X68000
日本語入力プロセッサー
FIXER ver.4.0
FPコール対応版

変換効率の良さで定評の高い“FIXER Ver.4.0”
X68000版がキーアサイン、FPコール対応により、
日本語ワープロ等より多くのアプリケーションで
御使用いただけるようになりました。

技術は夢から生まれる
Citysoft
シティソフト株式会社
〒534 大阪市都島区善源寺町2-7-5
TEL.06-927-1060 FAX.06-927-1067

※広告の内容は変更することがあります。

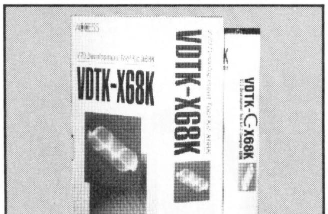
ACCESS for X68000

V70アクセラレータがどうして速いのか
そなただけにそっと教えてしんぜよう。
実はな、魔法のランプのように
魔人が住んでいるからなのじゃよ。
この魔人は、X68000の命令なら
何でも言う事を聞くのじゃが
特に芸術的な事ならなんでもござれの
凄いやつじゃ。
グラフィック演算でも、レイトレーシングでも
そなたが命令すればあっという間じゃ!
V70とX68000との面倒なやりとりも
全部やってくれるから
そなたはゆっくり茶でも飲んでいてくだされ。
おっと、そろそろ絵が描き上がる頃じゃ。
このことは決して他の者に教えるでないぞ……。

誰も知らなかった

V70のヒ・ミ・ツ

V70 アクセラレータ　VDTK-X68K



※製作：ボード……………有限会社アクセス
　　　　ソフトウェア………株式会社ハドソン

VDTK-X68Kの仕様

- ●V70 CPU(μPD70632)
  20MHz 32ビットマイクロプロセッサ
- ●V70AFPP(μPD72691)
  フローティング・ポイント・プロセッサ
- ●メインメモリ(DRAM)2Mバイト
  同一ページ内のアクセスはNo Wait
- ●共有メモリ(SRAM)128Kバイト
  X68000との通信用
- ●併行動作　X68000とV70は、併行に動作することが可能。
  データの受け渡し処理のために双方向ハンドシェークI/Oポートを搭載。

同梱ソフトウェア

- ●アセンブラ
- ●リンカ
- ●ソースコードデバッガ
- ●システムモニタ
- ●フロートエミュレータ
- ●コマンドシェル

オプションソフトウェア

- ●Cコンパイラ
  (VDTK-C-X68K)

価格

●ボードパッケージ　(XVI対応)
VDTK-X68K……………………**¥248,000**

●オプションソフト（Cコンパイラ）
VDTK-C-X68K…………………**¥68,000**

購入方法

上記商品は当面の間、通信販売のみとさせて頂きます。
購入ご希望の方は、住所、(社名、所属)氏名、電話番号をお知らせ下さい。注文書をお送りいたします。

有限会社アクセス
〒101 東京都千代田区神田神保町1-64
神保町協和ビル7F
☎03(3233)0200(代)　FAX.03(3291)7019

資料請求券
oh!X
12月号

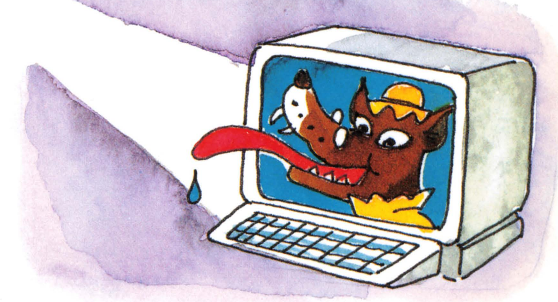
パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

タイムトラベルシリーズ

第9回

《もし、この時代にパソコン通信があったなら》



世間知らずの大胆さ

赤ずきんちゃん

赤いずきんがよく似合う赤ずきんちゃんは、とても可愛い女の子。ある日病気のおばあさんにワインとケーキを届けるために森の中へとお使いに出ます。でも、世間知らずの赤ずきんちゃんは、森の中で出会った狼の口車にのせられて、花つみをし、道草をしてしまいます。狼はその間におばあさんをひとのみ。遅れてやって来た赤ずきんちゃんもおばあさんに化けた狼に飲み込まれてしまいます。おなかのふくれた狼はそのままぐっすりいびきをかいて眠りこけます。通りかかった猟師はあまりに大きなイビキを不審に思い、部屋の中へ。猟師は狼を見つけ腹を裂き、赤ずきんちゃんとおばあさんを助けたのでした。

もし、この時代にパソコン通信があったなら。

「寄り道せずに、まっすぐにおばあさんの家にいくんですよ」赤ずきんちゃんはお母さんにそう言われたのに、つい狼にだまされて道草してしまいます。それもこれも赤ずきんちゃんが狼の怖さ、ずる賢さを知らなかったがため。もしこの時代にパソコン通信があったなら、出かける前に狼のことを教えてくれる人がいたかもしれません。パソコン通信はいわば部屋の中からのぞける「世間」。広い世間のあれやこれやをディスプレイの中から学べます。

パソコン通信なら、こんな楽しさ。

パソコン通信なら、何万人もの人たちが、自分の体験したことや知っていることをいつも語り合っています。そこはまさに生活感あふれる「世間」そのもの。実社会で学ぶのと同じくらい濃厚な体験が可能なパソコン通信の世界にあなたも飛び込んでみませんか？

きっと、出会える。キットで、会える。

買ったその日から2週間無料でアクセスできます。

J&P HOTLINEへのご入会はスタータキットで。お求めは、下記のお店でどうぞ。または現金書留にて、¥3,000+¥90（消費税3%）=¥3,090を、事務局までお送り下さい。すぐにスタータキットをお送りします。

お問い合わせは――――

〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOTLINE事務局宛 TEL(06) 632-2521

スタータキットのお求めはJ&P各店でどうぞ。

渋谷店	東京都渋谷区道玄坂2-28-4	☎(03)3496-4141
町田店	東京都町田市森野1-39-16	☎(0427)23-1313
八王子店	東京都八王子市旭町1-1八王子そごう7F	☎(0426)26-4141
立川店	東京都立川市幸町4-39-1	☎(0425)36-4141
三鷹店	東京都三鷹市野崎1-20-17	☎(0422)31-6251
横浜店	横浜市西区北幸2-9-5横浜HSビル1F	☎(045)313-6711
本厚木店	神奈川県厚木市中町3-4-4	☎(0462)25-5151
焼津インター店	静岡県焼津市越後島385	☎(054)626-3311
富山店	富山市掛尾町300	☎(0764)22-5033
金沢店	金沢市入江2-63	☎(0762)91-1130
寺地店	金沢市寺地2-3	☎(0762)47-2524
大須店	名古屋市中区大須4-2-48	☎(052)262-1141
テクノランド	大阪市浪速区日本橋5-6-7	☎(06) 634-1211
メディアランド	大阪市浪速区日本橋5-8-26	☎(06) 634-1511
コスモランド	大阪市浪速区難波中2-1-17	☎(06) 634-3111
U. S. LAND	大阪市浪速区日本橋4-9-15	☎(06) 634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06) 348-1881
高槻店	高槻市高槻町11-16	☎(0726)85-1212
くずは店	枚方市楠葉花園町15-2	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3 SENCHU PAL 2番街4F	☎(06) 834-4141
摂津富田店	高槻市大畑町24-10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4-20	☎(0720)34-1166
枚方バイパス店	枚方市田口3-41-7	☎(0720)48-1211
藤井寺店	藤井寺市岡2-1-33	☎(0729)38-2111
岸和田店	岸和田市土生町2451-3	☎(0724)37-1021
さんのみや1ばん館	神戸市中央区八幡通3-2-16	☎(078)231-2111
西宮店	西宮市河原町5-11	☎(0798)71-1171
伊丹店	伊丹市昆陽池1-63	☎(0727)77-5101
姫路店	姫路市東延末1-1住友生命姫路南ビル1F	☎(0792)22-1221
京都寺町店	京都市下京区寺町通仏光寺下ル恵比須之町549	☎(075)341-4411
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
和歌山店	和歌山市元寺町4-4	☎(0734)28-1441
和歌山南店	和歌山市中島368	☎(0734)25-1414
学園前店	奈良市学園北1-8-10	☎(0742)49-1411
奈良1ばん館	奈良市三条町478-1	☎(0742)27-1111
新大宮店	奈良市法華寺町83-5	☎(0742)35-2611
郡山インター店	大和郡山市横田693-1	☎(07435)9-2221
田原本店	奈良県磯城郡田原本町千代574-1	☎(07443)3-4041
熊本店	熊本市手取本町4-12	☎(096)359-7800

SHARP

目の付けどころが、
シャープでしょ。

いわば"感性"専用。

ことマインドに関しては
「汎用」という概念は存在しないも同じです。
「実用的である」のと、これなら「使える」というのも違います。
X68000が、普通のパソコンとは違うといわれる所以もここにあります。
いわゆる実用性を重視したビジネスパソコンとは
創造力で一線を画しています。
何に使うのか、何がしたいのか、
パソコン選びのポイントは目的にあったマシンを探すこと。
普通のパソコンに合わせるのでは
あなたのせっかくの創造力も発揮されません。
X68000は、使う人のクリエイティブマインドを咲かせる
"感性"専用パソコンです。

X68000
PERSONAL WORKSTATION・XVI
Compact

本体＋キーボード＋マウス
2HD3.5インチFDDタイプ CZ-674C-H(グレー) 標準価格298,000円(税別)
14型カラーディスプレイ(ドットピッチ0.28mm)
CZ-608D-H(グレー) 標準価格94,800円(税別)

- 5.25インチ増設用フロッピーディスクドライブ CZ-6FD5 標準価格99,800円・税別〔接続ケーブル同梱〕
- ディスプレイテレビ/CZ-6TU用RGBケーブル CZ-6CR1 標準価格4,500円・税別
- ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル CZ-6CT1 標準価格5,500円・税別
- SCSI変換ケーブル CZ-6CS1 標準価格12,000円・税別

(カラー液晶ディスプレイとの組み合わせ例)

10.4型TFTカラー液晶ディスプレイ
LC-10C1-H(グレー)標準価格598,000円(税別)
接続ケーブル AN-1515X 標準価格4,200円(税別)

※カラー液晶ディスプレイを接続してご使用の場合、SX-WINDOW上のアプリケーション利用に限定されます。

●お問い合わせは…
シャープ株式会社 電子機器事業本部システム機器営業部〒545大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表) 電子機器事業本部AVCシステム事業推進室〒162東京都新宿区市谷八幡町8番地☎(03)3260-1161(大代表)

T1002179120601 雑誌 02179-12

12月号 1992年12月1日発行(毎月1回1日発行) 通巻128号 昭和58年11月2日第三種郵便物認可 アスキー株式会社発行 Printed in Japan 定価800円(本体583円)